



Universidad  
Carlos III de Madrid

TRABAJO FIN DE GRADO

# DESARROLLO DE UN SISTEMA DE RECOMENDACIÓN DE INVERSIONES EN FUNCIÓN DE NECESIDADES DE LIQUIDEZ

Autor: Alejandro Dueñas Paz

Tutor: David Quintana Montero

Grado: Doble Grado en Ingeniería Informática y A.D.E.

Madrid, septiembre de 2017

## Resumen

El presente trabajo estudia la aplicabilidad de la computación evolutiva a la gestión de riesgos financieros. Concretamente se plantea el problema de cómo invertir el capital disponible de manera que se optimice la probabilidad de hacer frente a todos los flujos de caja negativos previstos. Mediante el uso de estrategias evolutivas se desarrolla una herramienta que de manera simplificada soluciona el problema planteado. Dados unos flujos de caja determinados y unos activos financieros disponibles, el algoritmo encuentra, en base a los datos históricos de estos activos financieros, la combinación óptima de los mismos para reducir el riesgo de falta de liquidez. En el trabajo se describe la herramienta y se muestran los resultados de algunas pruebas planteadas.

**Palabras clave:** Inteligencia Artificial. Computación Evolutiva. Riesgos Financieros. Inversión. Liquidez.

## Índice

Capítulo 1. Introducción.....	3
Introducción y objetivos.....	3
Contexto socioeconómico .....	5
Motivación .....	7
Capítulo 2. Revisión de la literatura.....	9
Introducción a la Computación Evolutiva.....	9
Estrategias Evolutivas .....	18
Aplicación en finanzas .....	19
ALM – Asset and Liability Management.....	21
Capítulo 3. Estructura y limitaciones .....	23
Planteamiento del problema y limitaciones.....	23
Modelado de la solución .....	26
Recopilación de datos .....	30
Capítulo 4. Estudio empírico.....	31
Descripción del código y su estructura.....	31
Pruebas .....	32
Evaluación de los resultados .....	49
Capítulo 5. Planificación, Presupuesto y Marco regulador .....	53
Planificación.....	53
Presupuesto .....	55
Marco regulador.....	56
Capítulo 6. Conclusiones.....	58
Conclusiones .....	58
Trabajo futuro.....	58
Impacto en el contexto socioeconómico .....	59
Bibliografía .....	61
Glosario .....	64
Anexos.....	66
English Summary .....	66
Abstract .....	66
Introduction and goals.....	67
Summary .....	71
Conclusions and future work.....	76

# Capítulo 1. Introducción

## Introducción y objetivos

La Inteligencia Artificial está cada vez más presente en todos los sectores. El incremento en la productividad que supone la aplicación de la IA es incuestionable. Según un informe desarrollado por Accenture, el capital y el trabajo, los hasta ahora motores del crecimiento económico, están pasando a un segundo plano frente a la IA, que está transformando la manera de generar crecimiento [1].

Una de las áreas afectadas por esta transformación que está provocando la IA en un creciente número de dominios es la de las finanzas. La constante aparición de nuevas fintechs (nombre que reciben las empresas dedicadas a la utilización de las tecnologías de la información para ofrecer servicios financieros) es una muestra de ello. Las instituciones financieras y otras compañías del sector están teniendo que adaptarse rápidamente a las nuevas tecnologías para mantener su posición en el mercado. Algunos ejemplos de la transformación que está provocando la IA en el sector son los nuevos sistemas de detección de fraude basado en IA, los chatbots bancarios o el trading algorítmico.

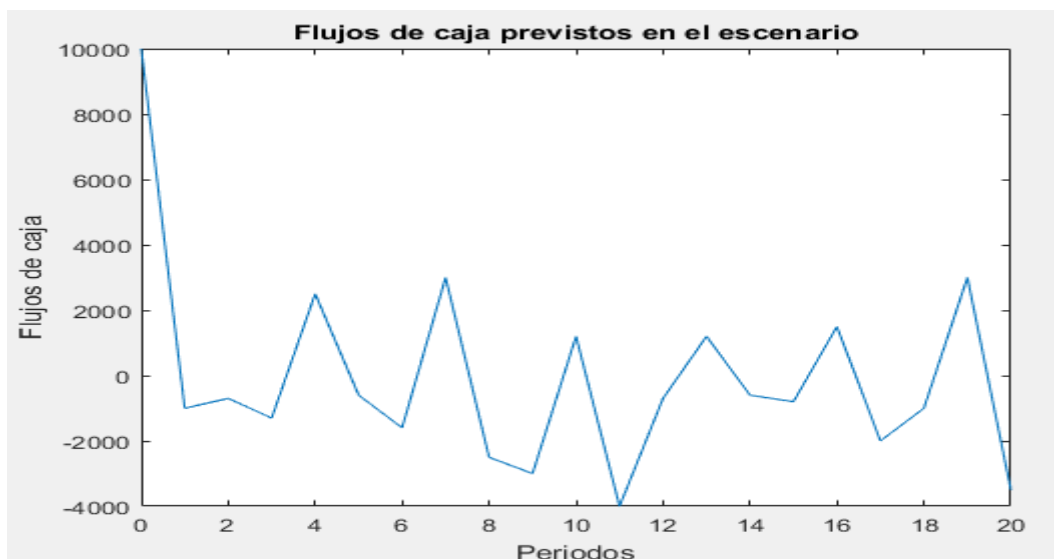
En este trabajo se muestra la utilidad y el valor que podría aportar la IA, en particular la computación evolutiva, en el campo de la gestión de riesgos financieros, desarrollando una herramienta con dicho objetivo.

En el presente trabajo se desarrolla un sistema de asignación de recursos financieros en función de requisitos de liquidez. El sistema está basado en la computación evolutiva, concretamente estrategias evolutivas. No obstante, se trata de una implementación simplificada, es decir, una prueba de concepto con el propósito de verificar la posibilidad de obtener resultados satisfactorios aplicando esta rama de la IA con este objetivo. No se pretende desarrollar una herramienta completamente funcional para uso profesional.

El problema subyacente al desarrollo de esta herramienta es como puede una compañía gestionar sus recursos financieros para minimizar el riesgo de liquidez, es decir, en que activos financieros distribuir su capital para maximizar la posibilidad de poder hacer frente a todos los pagos. Este problema es el motor para el desarrollo de la herramienta,

no obstante, puede ser usada con diferentes enfoques en el campo de las finanzas, no exclusivamente con el mencionado. Un ejemplo que se comentará en detalle más abajo es el de la planificación financiera personal frente a la empresarial ya mencionada. El enfoque en este caso sería como administrar los ahorros para lograr un determinado objetivo, por ejemplo: si tengo una capacidad de ahorro mensual de X euros, como invertirlos para tener con 60 años una cantidad Y que me permita jubilarme sin depender de un sistema público de pensiones. En ese supuesto durante todos los meses se dispondría de un flujo de caja positivo de X y finalmente con un flujo negativo de Y. El sistema recomendaría una distribución del capital en activos financieros que garantizase en un mayor número de casos poder disfrutar de esa cantidad Y en la fecha establecida.

En un escenario real, para realizar una ejecución profesional sería necesario contar con una base de datos muy completa que contase con datos históricos de activos financieros muy diversos: deuda de diferentes países, deuda de compañías en diferentes mercados, acciones de empresas con procedencia y actividad muy diversa, índices de todos los mercados, etc. Mientras más información estuviese disponible mejores serían los resultados. También habría que realizar un estudio más cuidadoso de las previsiones de flujos de caja, incluso elaborar un sistema de simulación del comportamiento de los activos financieros más completo que del que se hace uso en esta prueba de concepto. Sin embargo, el coste computacional de todo lo anterior sería elevadísimo y como ya se ha mencionado, el objetivo no es desarrollar una herramienta completa, sino probar su utilidad, valor y posibilidad de escala. Para esto se toman algunas simplificaciones y asunciones que se comentaran detalladamente más abajo.



**Figura 1.** Ejemplo de flujos de caja esperados para un caso.

A modo de ejemplo, en la Figura 1 se puede ver un gráfico con flujos de caja. Se comienza con una cantidad de 10.000 unidades monetarias y a lo largo de 20 periodos va entrando dinero (la gráfica está por encima de 0) y saliendo dinero (caso contrario). El trabajo consiste por lo tanto en el desarrollo de una herramienta que, dados unos activos financieros limitados (opciones de inversión) y unos requisitos de flujos de caja preestablecidos, sea capaz de definir una combinación de estos activos financieros que maximiza la probabilidad de hacer frente a todos los flujos de caja negativos. Es decir, que siempre que haya un flujo de cada negativo se disponga de capital para pagarlo. En el caso de la Figura 1 por ejemplo los flujos de caja negativos tienen un valor mayor que los positivos (teniendo en cuenta el capital inicial) por lo tanto en este caso si no se generan beneficios a través de la inversión del capital no se podrían realizar los pagos.

El funcionamiento del algoritmo consiste en: una simulación exhaustiva de los posibles escenarios futuros, es decir comportamientos que los activos pueden tener. Para posteriormente buscar la combinación de estos activos que en un mayor número de escenarios pueda cumplir con los flujos de caja preestablecidos.

El objetivo del trabajo consta de dos objetivos diferenciados:

1. Analizar la aplicabilidad de técnicas de computación evolutiva en las finanzas.
2. Realizar una prueba de concepto de una herramienta que aplicando técnicas de computación evolutiva en el apoyo de la gestión de riesgos financieros aporte valor a sus usuarios.

Adicionalmente, mencionar que al final del documento se dispone de un breve glosario con descripciones simples de los términos financieros utilizados en el documento. A los lectores no familiarizados con el vocabulario financiero se recomienda su lectura antes de continuar para facilitar así la comprensión del documento.

## Contexto socioeconómico

Continuando con la introducción, estas empresas dedicadas a la utilización de las tecnologías de la información para ofrecer servicios financieros denominadas fintech, están desarrollando herramientas digitales completamente disruptivas con los modelos de

negocios establecidos, a través de la creación de nuevas y más eficientes maneras de proveer estos servicios.

En el contexto de la inclusión financiera, las fintech tienen un gran potencial. Gracias a la competitividad creada por sus nuevas herramientas tecnológicas desarrolladas los antiguos modelos de negocio se están viendo desafiados y teniéndose que adaptar rápidamente puesto que se está demostrando que los servicios financieros pueden ser provistos de manera mucho más rápida, responsables y eficiente.

El acceso a productos y servicios financieros es más alcanzable que nunca, especialmente para los consumidores que viven en regiones rurales sin estructuras propias de una economía moderna. Las fintech no pueden simplemente hacer esos productos y servicios más accesibles, también pueden facilitar y hacer más rentables los negocios a través de la reducción del coste de hacer negocios.

La inclusión en finanzas está creciendo rápidamente gracias a las fintech. Algunos de los avances más destacados que están logrando son: las microfinanzas, las aplicaciones electrónicas de servicios bancarios y préstamos permiten agilizar las decisiones de crédito y permiten a los prestamistas servir a más clientes y en diferentes lugares de manera simultánea. Transferencias monetarias internacionales, gracias a las fintech cada vez estas son más baratas y rápidas. Acceso a los historiales de crédito, existe mucha información financiera personal hasta ahora difícil de almacenar y compartir, lo que se traducía en complicar el acceso a crédito, las fintech facilitan la gestión de toda esta información. Pagos electrónicos, etc.

A modo de resumen, todo lo comentado parece guiarnos hacia una economía mucho más globalizada. Una economía en la que gracias a la tecnología será mucho más sencillo el acceso a servicios financieros por parte de cualquier persona sin importar su lugar de nacimiento. Esto redundará en un incremento de la facilidad de hacer negocios. El acceso a crédito será mucho más sencillo gracias a la mayor facilidad para acceder a historiales de crédito que unido a la simplificación de las transferencias internacionales resultará en un incremento en el emprendimiento, el cual es el motor en la generación de riqueza.

La pregunta a continuación sería, ¿cómo encaja la herramienta planteada dentro de este contexto disruptivo? Desde el enfoque de esta evolución a una economía más global e inclusiva, esta herramienta colaboraría en la gestión de riesgos financieros de manera eficiente en compañías que no puedan dedicar mucho capital a desarrollar un

departamento competitivo en este aspecto. Por otro lado, supondría una herramienta de apoyo con un gran aporte de valor que colabore en la toma de decisiones de compañías de mayor tamaño. Por último, enfocando la herramienta desde una perspectiva de apoyo en la toma de decisiones de inversión personales, podría apoyar a grandes volúmenes de pequeños inversores a un menor precio, colaborando así con la inclusión de más personas en los servicios financieros, al no necesitar grandes capitales para contratar a un gestor personalizado.

## Motivación

A nivel personal existen dos motivos principales que me han llevado a la elección de este tema como trabajo fin de grado. En un primer lugar, mis motivaciones para desarrollar una herramienta computacional aplicada a las finanzas. En un segundo lugar, por qué el uso de la computación evolutiva para desarrollar esta herramienta.

En primer lugar, como estudiante del doble grado en Ingeniería Informática y Administración de Empresas la formación que he recibido ha sido híbrida, siendo las finanzas la parte que más me ha interesado por la parte de A.D.E. y la computación por parte de la informática. Con este trabajo intento hacer uso de técnicas de inteligencia artificial con un enfoque de apoyo en la toma de decisiones financieras. Aunque se trate de un trabajo exclusivo de informática, no uno mixto de ambos grados, me motiva mucho más el desarrollo de una herramienta a la que le vea una utilidad práctica en un campo que me gusta como es las finanzas, que un desarrollo más teórico o aplicado en otro campo que me resulte menos interesante.

En segundo lugar, la elección de hacer uso de la computación evolutiva está motivada por el hecho de que en mi plan de estudios (plan de 2011) es imposible recibir formación en esta materia. El plan de estudios nos deja muy poca libertad para configurarnos asignaturas libremente, permitiéndose tan solo elegir entre la mención en Sistemas de Información o en Computación, no incluyendo esta última ninguna asignatura centrada en la computación evolutiva. Por las menciones que se realizaban sobre la computación evolutiva en otras asignaturas más genéricas como Inteligencia artificial en las organizaciones, me parecía un tema muy interesante y para completar mi formación en



esta rama decidí aplicarla en el desarrollo de mi trabajo fin de grado de Ingeniería Informática.

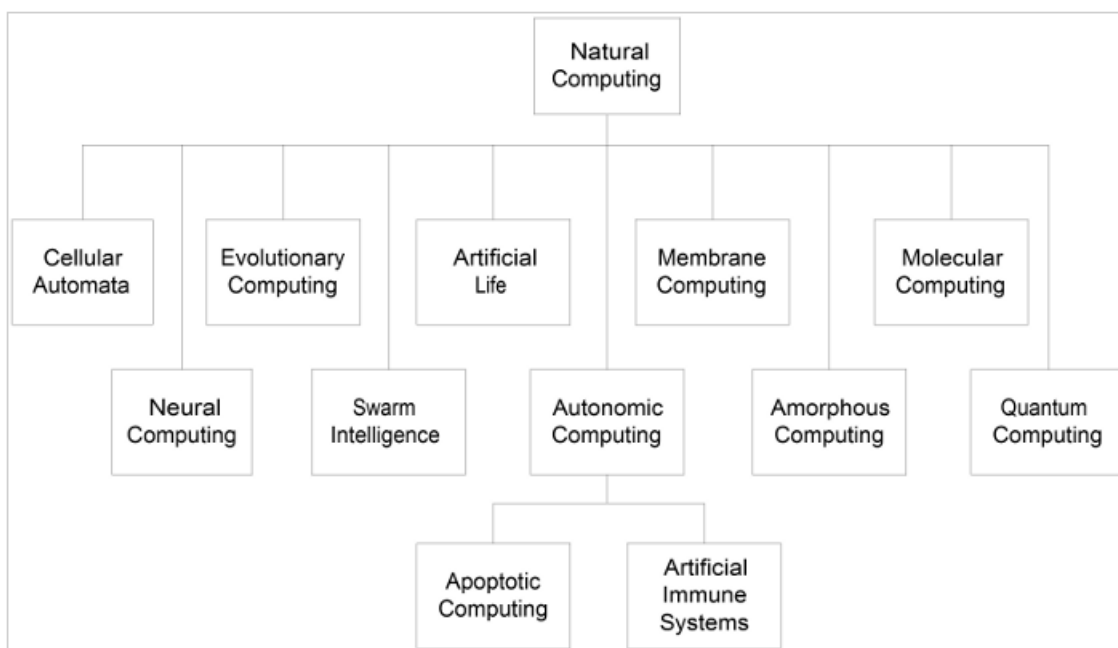
Por otra parte, el contexto socioeconómico ha sido otro factor que ha propiciado mi elección de este tema. Como se ha descrito en el apartado anterior, la inteligencia artificial está generando cambios muy significativos que afectan a prácticamente cualquiera aspecto de la vida. La inclusión de estas tecnologías está suponiendo un incremento de la productividad sin precedentes, lo que se traduce en un aumento de la riqueza que va a cambiar la manera de ver el mundo a nivel global. Por lo tanto, ¿qué mejor manera para contribuir a esta evolución hacia una economía cada vez más rápida, eficiente y globalizada que participando del desarrollo de herramientas con tal objetivo?

## Capítulo 2. Revisión de la literatura

### Introducción a la Computación Evolutiva

La Computación Natural o No Convencional (*Natural Computing*) puede ser definida de manera general como el desarrollo de programas informáticos y algoritmos computacionales usando inspiración de sistemas y fenómenos que ocurren en la naturaleza.

Los algoritmos de Computación Natural pueden ser agrupados en diferentes grupos dependiendo de en que aspectos del mundo real están basados. Algunos de estos grupos son: *Neural Computing*, *Evolutionary Computing*, *Autonomic Computing*... (ver Figura 2) [2].



**Figura 2.** Clasificación Computación Natural [2].

La Computación Evolutiva (*Evolutionary Computation*) está basada en los principios Darwinianos de la evolución. Se trata de un enfoque de resolución de problemas centrado en una población en la que múltiples soluciones candidatas son tratadas en paralelo. Se usan operadores de búsqueda genéticos para identificar las soluciones (individuos de la población) más capaces de resolver el problema (las mejor adaptadas) y las siguientes

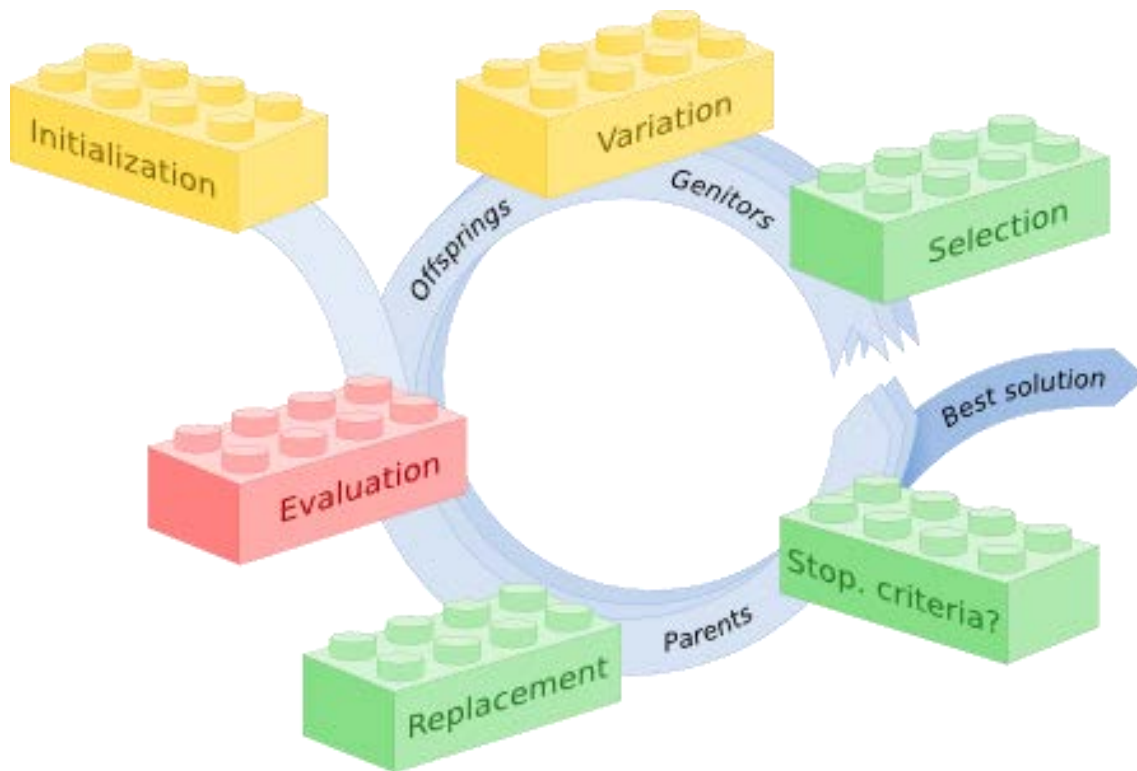
generaciones son creadas usando una selección basada en la “adaptación” de estas. La “adaptación” (*fitness*) de una solución candidata es una medida de su capacidad para resolver el problema.

Un creciente número de investigadores están buscando aplicaciones de métodos de Computación Natural en finanzas, en particular de la Computación evolutiva. Esto se refleja en el significativo aumento de conferencias, talleres, etc. en la materia. Ejemplo de esto es la conferencia anual sobre Computación Evolutiva en Finanzas y Economía celebrada en el congreso del IEEE sobre Computación Evolutiva o la Conferencia anual internacional sobre Inteligencia Computacional en Economía y Finanzas (Computational Intelligence in Economics & Finance CIEF) [3].

Desde que Charles Darwin popularizó la teoría de la Selección Natural, la fuerza motriz detrás de la evolución, la biología molecular ha descifrado alguno de los misterios que no hacían teorías anteriores, ejemplo de esto es la existencia y estructura del ADN. El Neo-Darwinismo, que representa la acumulación de conocimiento sobre los procesos de la evolución a nivel molecular junto con la teoría de la evolución de Darwin, ha dado lugar a una familia de algoritmos de resolución de problemas conocidos como Computación Evolutiva.

El origen de la Computación Evolutiva se puede rastrear desde al menos el origen de la informática [4] con los escritos de Turing [5] donde discute la posibilidad de una búsqueda genética o evolutiva (“*Genetical or evolutionary search*”) como uno de los elementos claves en los que podría basarse la inteligencia de las máquinas (“*the intelligence of machinery*”). Existen implementaciones de Computación Evolutiva que datan de la década de los 50 con los pioneros de esta que popularizaron sus ideas en décadas posteriores, esto incluye a Holland [6], Fogel, Owens and Walsh [7], Rechenberg [8], Schwefel [9], Goldberg [10], D Jong [11], and Koza [12].

El proceso natural de evolución trata positiva o negativamente a las especies según su éxito a la hora de sobrevivir y reproducirse en su entorno. Los diferentes supervivientes y la variedad generada durante la reproducción forman el motor de la evolución [13], [14]. La Computación Evolutiva implementa este proceso como se muestra en la Figura 3.



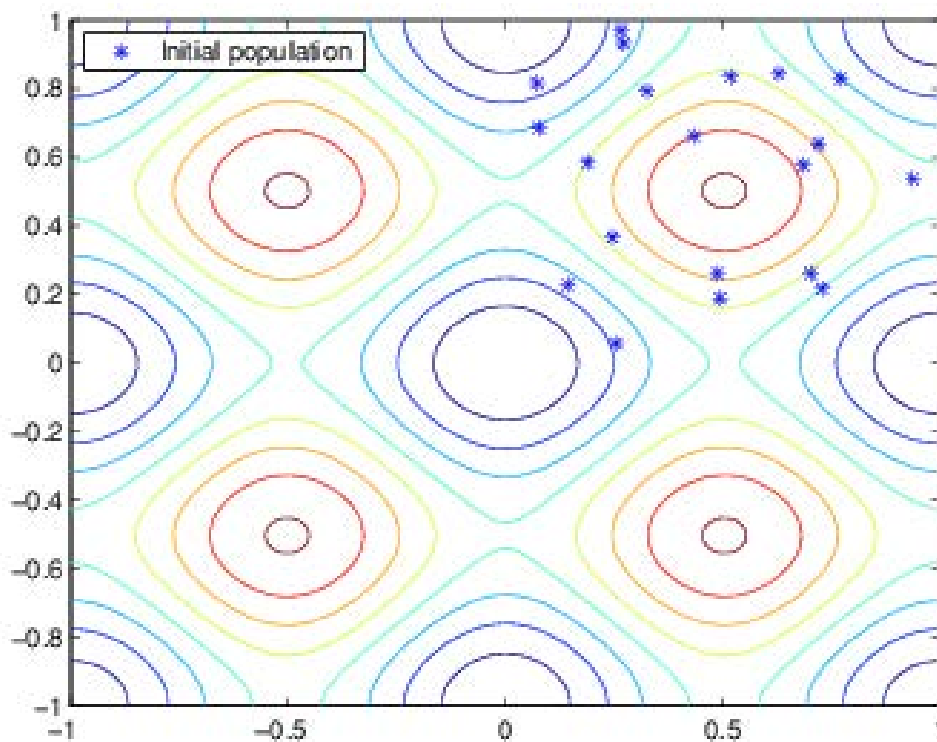
**Figura 3.** Proceso implementado en Computación Evolutiva [15].

En Computación Evolutiva, el equivalente al ADN es la representación o codificación que adopta cada miembro de la población (habitualmente llamados individuo o solución candidata) para resolver un problema específico. Esta representación de los individuos puede ir desde una simple cadena de longitud fija de números binarios o reales. Hasta estructuras más complejas como gráficos o incluso fragmentos de código de longitud variable que pueden ser ejecutados.

La estructura que sigue un algoritmo genético es la siguiente [16]:

1. El algoritmo comienza con la creación de una población inicial aleatoria. A veces esta población inicial parte con soluciones candidatas heurísticas, pero es importante que el grueso de la población sea de generación aleatoria para no converger a un óptimo local de manera prematura.
2. El algoritmo genera las siguientes poblaciones de soluciones candidatas. En cada paso el algoritmo utiliza individuos de la generación actual para crear la siguiente población. Para generar esta nueva población, el algoritmo realiza los siguientes pasos:

- a. Asigna a cada miembro de la población actual un valor de adaptación que cuantifica cuan buena es la solución.
  - b. Selecciona individuos de la generación actual, llamados padres, en base a su valor de adaptación. A mayor valor de adaptación más probabilidad de ser seleccionado, pero todos los individuos deben tener opción a ser seleccionados para, igual que antes, evitar llegar a un óptimo local prematuramente.
  - c. Alguno o algunos de los individuos pueden ser seleccionados para que pasen directamente a la siguiente generación, a esto se le conoce como elitismo.
  - d. Se producen hijos de los individuos seleccionados como padres. Las soluciones hijo pueden ser producidas haciendo cambios aleatorios en un solo padre (mutación) o combinando a un par de padres (cruce).
  - e. Se sustituye la generación actual por los hijos generados (más los individuos elite si existen).
3. El algoritmo para cuando se cumple la condición de parada.



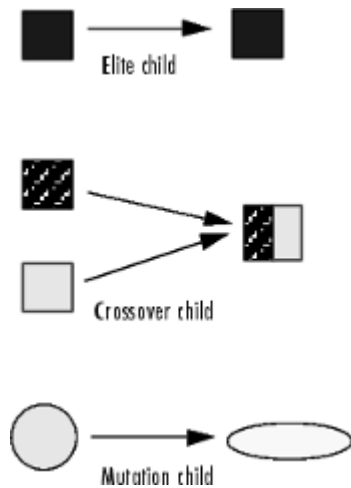
**Figura 4.** Ejemplo de representación gráfica de una población inicial.

El algoritmo comienza creando una población inicial aleatoria como se muestra en la Figura 4. En este ejemplo la población inicial contiene 20 individuos. En este ejemplo todos los individuos iniciales aparecen en el cuadrante superior derecho, es decir, entre las coordenadas 0 y 1. Esto podría provocar que la solución convergiese en un óptimo local.

En cada generación el algoritmo usa la población para generar los hijos que formaran la siguiente generación. El algoritmo selecciona a un grupo de individuos de la población, llamados padres, que contribuyen con sus genes (las entradas de su vector) a sus hijos. El algoritmo suele seleccionar individuos que tienen mejores valores de adaptación que sus padres. La función que se usa para seleccionar a los padres es habitualmente selección por ruleta o selección por torneo. La selección por ruleta elige a los padres de manera aleatoria otorgando más probabilidad de ser elegido mientras mejor valor de adaptación tiene. La selección por torneo, la usada en este trabajo, toma X miembros de la población con igual probabilidad y posteriormente selecciona el individuo con mejor valor de adaptación entre estos.

El algoritmo crea tres tipos de hijos para a siguiente generación, ver Figura 5:

- Los individuos elite son las soluciones candidatas que pasan directamente a la siguiente generación por su mejor valor de adaptación, se ha demostrado que un algoritmo genético requiere elitismo para poder converger al óptimo global [17]. En el trabajo usaremos un elitismo de un individuo, la mejor solución candidata de cada generación pasará directamente a la siguiente.
- Los individuos hijos por cruce son los generados por la combinación de los vectores de dos padres.
- Los individuos hijos mutados son creados mediante la introducción de cambios aleatorios (mutaciones) en un solo padre.

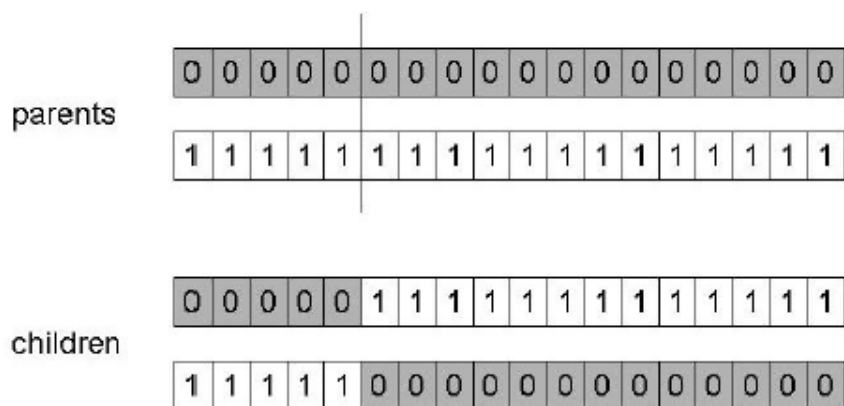


**Figura 5.** Representación gráfica de los hijos generados por un algoritmo genético.

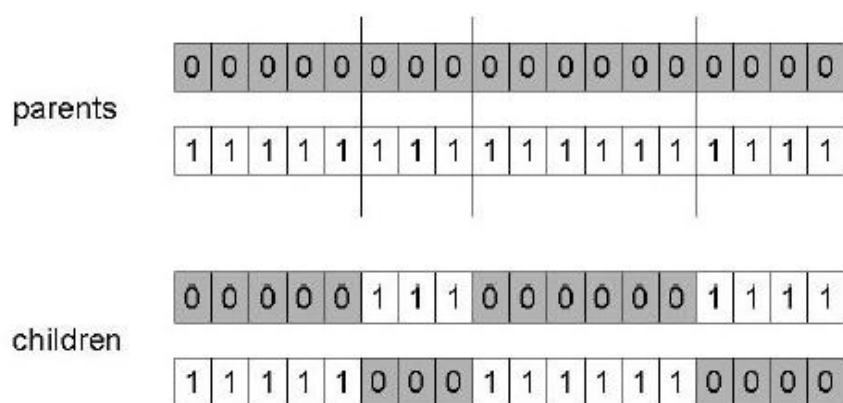
Los operadores genéticos nos permiten generar nuevas soluciones candidatas. Se suelen clasificar según su aridad (esto es, de su número de entradas). La selección de operadores concretos depende del mecanismo de representación escogido [18] - [23].

- **Mutación** (aridad 1). Los operadores de mutación actúan sobre el genotipo, es decir, el vector. Es esencial su aleatoriedad, esto es lo que los diferencia de operadores heurísticos unarios. La mutación introduce diversidad en la población. La mutación estándar altera el valor de cada gen, es decir, de cada entrada del vector individuo, con una probabilidad  $p$ . A esta  $p$  se le llama probabilidad de mutación.
- **Cruce** (aridad 2). Combinan información de los padres para crear nuevos descendientes. La selección de que información de los padres se combina es estocástica. Puede provocar que muchos descendientes sean peores que los padres en términos de la función de adaptación. Existen varios tipos de cruces:
  - **Cruce en un punto.** Se selecciona un punto de cruce aleatoriamente (una posición del vector padre). Se dividen los padres en ese punto y se crean los hijos intercambiando los vectores, ver Figura 6. Los problemas de este tipo de cruce es que hace más probable que continúen juntos genes que están más cerca (posiciones del vector más cercanas) además de nunca mantener juntos genes de extremos opuestos.
  - **Cruce en  $n$  puntos.** Se eligen  $n$  puntos de cruce aleatoriamente, se fragmentan los cromosomas en esos puntos y se juntan los fragmentos alternando a los padres, ver Figura 7.

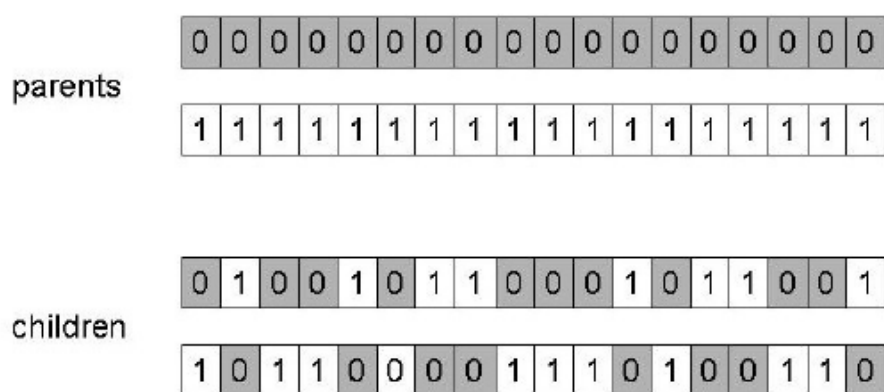
- **Cruce uniforme.** Se elige al azar el padre del que proviene cada gen (posición del vector) para el primer hijo y el segundo hijo es su inverso, ver Figura 8.



**Figura 6.** Ejemplo gráfico de cruce en un punto.



**Figura 7.** Ejemplo gráfico de cruce en n puntos.

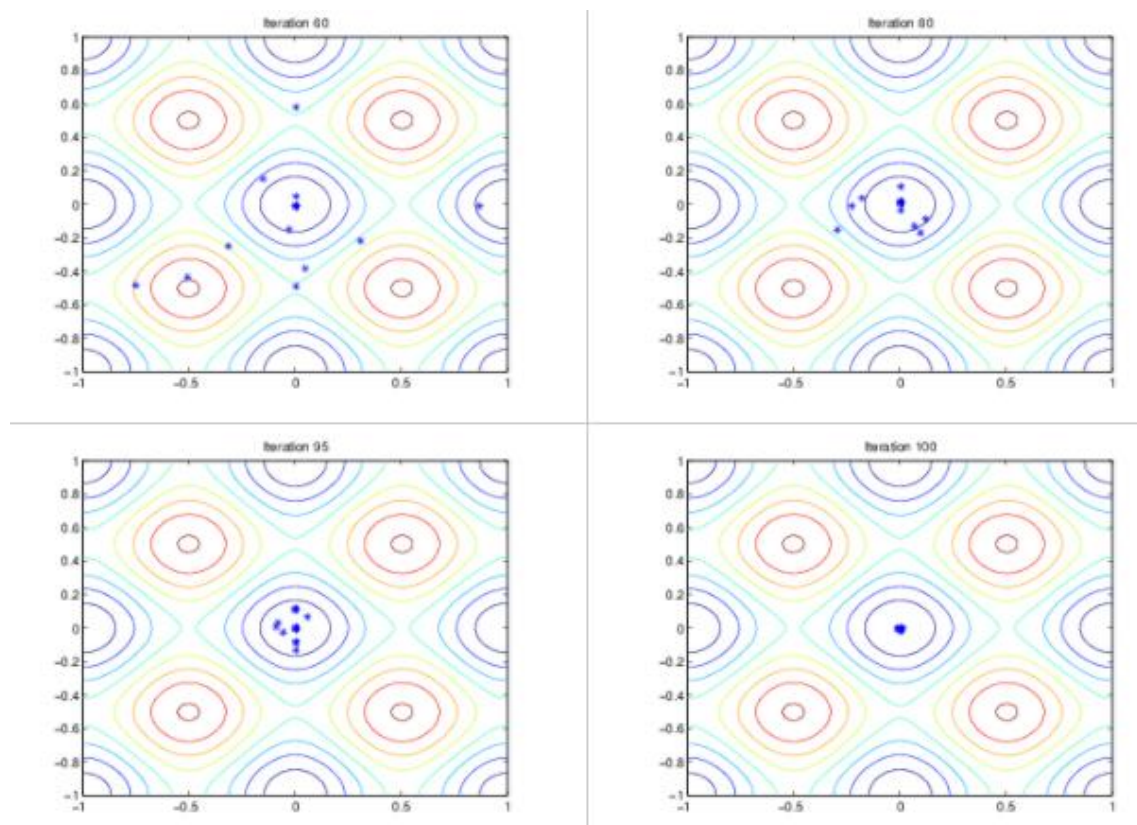


**Figura 8.** Ejemplo gráfico de cruce uniforme.



¿Cruce o mutación? Depende del problema, pero en general suele ser bueno utilizar ambos operadores. El operador de cruce explora el espacio de búsqueda, descubre áreas prometedoras lejos de los padres. El operador de mutación explota lo que ya hemos descubierto, permanece cerca del área de los padres. Con el cruce se puede combinar lo mejor de dos padres, sin embargo, la mutación permite introducir nuevas entradas en los vectores individuo (el cruce no cambia la proporción de cada entrada del vector en la población). En el desarrollo de esta herramienta se hará uso de ambos operadores como se mostrará más abajo.

La Figura 9 muestra la población en las generaciones 60, 80, 95 y 100 que siguen a la inicial mostrada anteriormente en la Figura 4. Se puede ver como a medida que aumenta el número de generaciones, los individuos de la población se acercan unos a otros y se aproximan al punto óptimo, el  $[0,0]$ .



**Figura 9.** Ejemplo gráfico de la evolución de las poblaciones en las diferentes generaciones.

Hay varias maneras de determinar la parada del algoritmo. Algunos criterios típicamente usados son: el número de generaciones, tras un determinado número de generaciones el algoritmo para. Un límite de tiempo, el algoritmo para después de ejecutar durante un determinado periodo de tiempo. Nivel de adaptación, el algoritmo se detiene cuando el

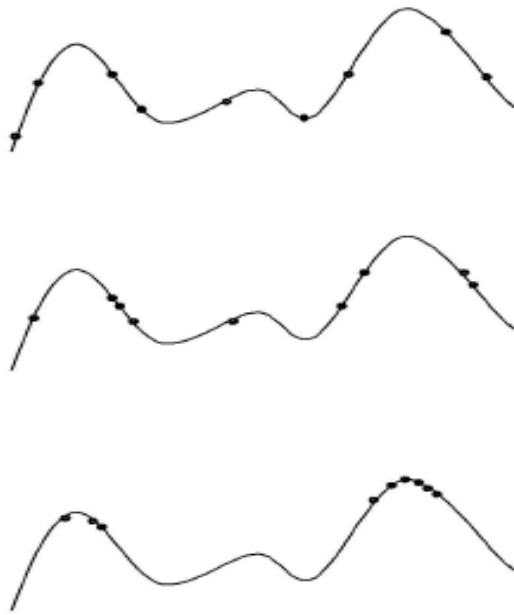
valor de adaptación del mejor individuo de una población alcanza un límite preestablecido. Otros criterios de parada hacen referencia a los incrementos en la función fitness, por ejemplo.

La función de selección elige padres para las siguientes generaciones basándose en sus valores de adaptación determinados por la función de fitness. Un individuo puede ser seleccionado como padre en más de una ocasión, en cuyo caso contribuiría en la construcción de más de un hijo. En la selección suelen emplearse mecanismos estocásticos: es más probable que un mejor individuo sea seleccionado, pero nada lo garantiza, incluso el peor individuo de la población puede ser seleccionado. Esta selección estocástica nos ayuda a escapar de óptimos locales.

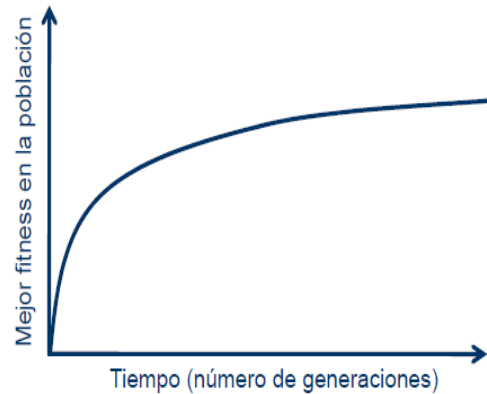
- **Selección proporcional** (“ruleta”). La idea de este mecanismo es que los mejores individuos tienen más posibilidades de reproducirse (proporcionalmente a su nivel de adaptación). La implementación trataría de asignar a cada individuo una parte de la ruleta proporcional a su fitness y posteriormente “girar” la ruleta para realizar la selección de la población.
- **Selección basada en el ranking.** Para eliminar los problemas de la selección proporcional, las probabilidades de selección se basan en valores relativos en vez de en los valores absolutos de fitness. Se ordena la población de acuerdo con su fitness y las probabilidades de selección se basan en el ranking (se trata de una idea similar a los test no paramétricos en estadística). Puede ser un ranking lineal o un ranking exponencial.
- **Selección por torneo.** La idea básica es escoger  $n$  miembros de manera aleatoria y seleccionar de ellos el que mejor valor de fitness tenga repitiendo el proceso las veces necesarias. La probabilidad de seleccionar a un individuo dependerá del valor de adaptación del individuo, del número de miembros  $n$  elegidos al azar (a mayor  $n$  mayor presión en la selección) y el método de muestreo (con o sin remplazo).

En la Figura 10 vemos las fases por las que debería pasar la ejecución de un algoritmo evolutivo “ideal”. En primer lugar, las generaciones iniciales siguen una distribución aleatoria. Las generaciones intermedias comienzan a localizarse cerca de las elevaciones que representan los óptimos locales. Por último, las últimas generaciones se concentran en las cimas de las elevaciones mayores, los óptimos.

En la Figura 11 se ve la evolución típica del fitness, nivel de adaptación, de las generaciones a lo largo de la ejecución del algoritmo.



**Figura 10.** Ejemplo de evolución de una población a lo largo de las diferentes generaciones.



**Figura 11.** Ejemplo de la evolución del fitness de una población a lo largo de las diferentes generaciones.

## Estrategias Evolutivas

Las Estrategias Evolutivas (*Evolution Strategy*) son una técnica de optimización basadas en la teoría de la evolución. Estas son una rama de la Computación Evolutiva.

Estas técnicas de optimización fueron desarrolladas a principios de la década de los 60 y especialmente a partir de la década siguiente principalmente por Ingo Rechenberg y Hans-Paul Schwefel. Su objetivo inicial era resolver problemas hidrodinámicos de alto grado de complejidad [24].

Las Estrategias Evolutivas hacen uso de la mutación y selección como operadores principales de búsqueda. Estos operadores son aplicados de manera iterativa y cada una de estas iteraciones es denominada generación. La secuencia de generaciones continua hasta un determinado criterio de parada.

La mutación se realiza habitualmente añadiendo valores aleatorios a las cadenas que forman los individuos. La selección en Estrategias Evolutivas es determinista y está basada exclusivamente en el nivel de adaptación de los individuos frente al resto. Por lo tanto, el algoritmo resultante es independiente de transformaciones monótonas o de la función objetivo.

La versión original y más simple usaba un solo padre que generaba un solo hijo, es decir una población de dos. Este hijo se convertía en el padre de la siguiente generación si era mejor que el padre, en caso contrario era eliminado. A este tipo de selección se la conoce como extintiva, debido a que los individuos peor adaptados tienen una probabilidad nula de ser seleccionados. Esto sería  $(1 + 1)$  - ES. En 1973 Ingo Rechenberg introdujo el concepto de población, proponiendo una estrategia en la que un número fijo de padres ( $\mu$ ) genera un solo hijo, el cual puede reemplazar al peor padre de la población (selección extinta). Esto sería  $(\mu + 1)$  - ES. Una  $(1 + \lambda)$  - ES trataría de un solo padre que genera  $\lambda$  hijos los cuales compiten por ser el padre en la siguiente generación [25], [26].

En la actualidad las Estrategias Evolutivas suele usar una población de  $\mu$  padres y hacen uso de un operador adicional, la recombinación [27]. Los operadores de combinación pueden ser: sexuales, el operador actúa sobre los individuos elegidos aleatoriamente de la población de padres. O pueden ser panmícticos, se elige un solo padre al azar, y se mantiene fijo mientras se elige al azar un segundo padre (de entre toda la población) para cada componente de su vector.

## Aplicación en finanzas

Las técnicas de computación se expanden rápidamente al mundo de las finanzas. Solamente en el área del trading algorítmico, Aite Group estima que en 2010 el 50% de las ordenes en Estados Unidos, el 28% en Europa y el 16% en Asia se ejecutaban de manera automática a través de algoritmos de trading [28]. Esto significa que más de 8,5 billones (miles de millones) de acciones se operan diariamente en EE. UU. de manera automática, es decir, más de 120 billones (miles de millones) de dólares en términos monetarios.

El modelado de los mercados financieros es complicado por diversas razones. Muchos factores afectan a los mercados, como son: las tasas de interés, las ratios de crecimiento de la economía o la liquidez. No existe una teoría que demuestre con exactitud cómo estos y otros factores afectan a los precios de los activos financieros. Otros aspectos que hacen el modelado de los mercados financieros complicado son: la ausencia de historiales de muchos de estos con los que contrastar los modelos teóricos; la naturaleza emergente de estos; y su inherente imprevisibilidad debido a factores que pueden afectarlos como desastres naturales. A pesar de todas estas dificultades muchos inversores se suman al uso de técnicas computacionales. En los últimos años especialmente el uso de la Computación Natural ha atraído significativamente el interés de muchos inversores.

Las aplicaciones de la Computación Evolutiva en finanzas pueden ser clasificada como de optimización o de modelado (existe una amplia literatura sobre modelado basado en agentes en finanzas, pero no serán tratadas en este documento).

Las aplicaciones en casos de optimización, como selección de una cartera, abundan en finanzas. A menudo, debido a que el problema financiero que se intenta resolver se ve influido por un alto muy número de variables. Las metodologías de optimización que dan soluciones a este tipo de problemas son generalmente heurísticas como: Algoritmos Genéricos, Estrategias Evolutivas, Evolución Diferencial y algunas optimizaciones de colmena (*Swarm Optimization*).

La optimización en finanzas es importante, pero en ocasiones el modelo subyacente es desconocido. Por lo tanto, a menudo la tarea es identificar un modelo explicativo del conjunto de datos del que se dispone. Esta es una tarea complicada puesto que tanto la estructura del modelo como los parámetros relevantes deben ser descubiertos.

Desde finales de la década de los 80 se han realizado muchos estudios en finanzas desde una óptica evolutiva. Inicialmente se centraban en la aplicación de Algoritmos Genéticos para la optimización de parámetros y selección de variables [29] pero a partir de mediados de la década de los 90 se popularizó el uso de la Programación Genética en predicción financiera y sistemas de trading. En los últimos años, con la sofisticación de las técnicas se está aumentando su uso en finanzas, expandiéndose a otras áreas como predicción de la volatilidad.

Un excelente punto de partida en la investigación sobre aplicaciones de metodologías de Computación Evolutiva en finanzas, en áreas como la predicción financiera, evaluación

del riesgo de crédito, optimización de carteras, valoración de activos o trading algorítmico se puede encontrar en [30] - [34].

Uno de los problemas clásicos de optimización multiobjetivo son los de optimización de cartera, donde el objetivo es invertir una cantidad de dinero en diferentes activos (una cartera) con el objetivo de maximizar los beneficios mientras se minimiza el riesgo. El problema tratado en este trabajo es una variante de este problema en la que el objetivo no es maximizar los beneficios de la inversión, sino que estos sean suficientes para cubrir las exigencias de capital en el mayor número de posibles escenarios.

## ALM – Asset and Liability Management

La Gestión de Activos y Pasivos (ALM - Asset and Liability Management) es la gestión de los riesgos derivados de desajustes entre activos y pasivos. Los promotores de esta fueron las instituciones financieras durante la década de los 70 debido al aumento de la volatilidad en las tasas de interés.

El proceso es una combinación de la gestión de riesgos y la planificación estratégica. No se trata simplemente de ofrecer soluciones para mitigar o protegerse del riesgo que surge de la interacción de activos y pasivos, sino que se centra en una perspectiva largoplacista: ser capaz de maximizar los activos cumpliendo con los requisitos de los pasivos para así lograr un aumento de la rentabilidad.

Los sistemas modernos de ALM incluyen la gestión de activos, acciones, tasas de interés y gestión del riesgo de crédito. También establecen un marco desde el cual gestionar el riesgo teniendo en cuenta la regulación pertinente y el entorno económico.

La función exacta del ALM puede variar mucho de una compañía a otra dependiendo del modelo de negocio adoptado y puede abarcar la gestión de un área de riesgos mayor o menor.

Los sistemas tradicionales de ALM se centran en el riesgo de la tasa de interés y en el riesgo de liquidez debido a que son los que mayor efecto tienen en el balance de una compañía, además requieren de coordinación entre activos y pasivos.

Los sistemas de ALM actuales buscan un uso de más información, incluyendo por ejemplo los riesgos derivados del cambio de divisa y la gestión del capital. De acuerdo con una encuesta realizada por la auditora y consultora PricewaterhouseCoopers (PwC), el 51% de las 43 instituciones financieras líderes consultadas se preocupan de la gestión del capital en sus departamentos de ALM [35]. Este trabajo trata esta división del ALM, la gestión del capital, principalmente junto con la gestión del riesgo de liquidez, aportando un enfoque computacional para su optimización.

El alcance de las funciones de ALM cubren entonces los siguientes procesos:

- **Riesgo de liquidez.** El riesgo actual y potencial de que la compañía no pueda cumplir con sus obligaciones (deuda) a su vencimiento sin afectar adversamente las condiciones financieras de la misma. Desde una perspectiva ALM, la atención se centra en el riesgo de liquidez de financiación de la compañía, es decir, su capacidad para cumplir con sus obligaciones de flujos de efectivo actuales y futuros.
- **Riesgo de interés.** El riesgo de pérdidas derivadas de los movimientos de las tasas de interés y su impacto en los flujos de caja futuros.
- **Riesgo de mercado.** Riesgo asociado a las pérdidas del valor de un activo asociado a la fluctuación de su precio en el mercado.
- **Riesgo de cambio.** El riesgo de sufrir pérdidas derivadas de los movimientos en las tasas de cambio de divisas.
- **Obtención de fondos y gestión del capital,** entendido como mecanismo que asegura el mantenimiento de un nivel de capital adecuado. Es un proceso dinámico y continuo que tiene en cuenta las necesidades de capital a corto y a largo plazo y coordina los ciclos de planificación y la estrategia global (habitualmente con un horizonte temporal de dos años).
- **Planificación de resultados y crecimiento.**
- Adicionalmente, ALM trabaja con aspectos relacionados con el riesgo de crédito y como puede afectar a la cartera de crédito (inversiones, deuda y tesorería) en el balance. El riesgo de crédito, especialmente en la cartera de deuda, se gestiona con herramientas que representan una de las principales contribuciones de la computación al ALM.

## Capítulo 3. Estructura y limitaciones

### Planteamiento del problema y limitaciones

En primer lugar, es importante recalcar que en el presente trabajo se realiza una prueba de concepto, es decir, se intenta demostrar que la Computación Evolutiva puede ser aplicada en el campo de ALM (gestión del activo y pasivo) obteniendo resultados que aportan valor. En ningún momento se pretende desarrollar una herramienta completamente funcional para uso profesional, por lo tanto, el problema planteado es una simplificación de un caso profesional real.

Para el desarrollo de la herramienta se toman por lo tanto algunas simplificaciones y asunciones que facilitan su desarrollo y especialmente reducen su coste computacional. Estas limitaciones impuestas a la herramienta la hacen testeable sin afectar a su escalabilidad, que con un mayor tiempo de desarrollo y máquinas capaces de soportar el coste computacional podría ser funcional en casos reales. Las limitaciones de la herramienta son:

- **Totalidad del capital invertido.** Se asume que inicialmente se invertirá todo el capital disponible y posteriormente todo fruto del capital invertido más los flujos de caja positivos de los que se vaya disfrutando se reinvertirán siguiendo la estructura de la cartera, es decir, no se reservará nada del capital en tesorería. Es muy poco probable que en un caso real no se mantenga una parte del capital en tesorería, este tipo de decisiones deberían ser tomadas de acuerdo con el cliente, tomando cada proyecto como un caso diferente y dándole un trato personalizado. Cada cliente debería elegir entre que márgenes de liquidez quiere mantenerse y la herramienta podría recomendar mantenerse fuera del mercado, o no invertir el 100% del capital en determinadas situaciones por ser más prudente, pero para esta prueba de concepto se elige este criterio que hace el problema menos complejo sin afectar a su escalabilidad.
- **Supuesto de reinversión según una estructura de capital fija.** Se asume que la estructura de capital de la cartera se mantiene fija en el tiempo. En el instante inicial se destinarían diferentes porcentajes del capital a comprar ciertos activos



financieros, cada uno de ellos a un precio determinado por el mercado en ese instante. Posteriormente, siguiendo la evolución natural del mercado, sus cotizaciones variarían provocando que los pesos fijados inicialmente en cada activo variasen, pasando los activos con un mejor comportamiento a tener un mayor peso y los de un peor comportamiento a tener un menor peso en la cartera. La simplificación que se realiza para la prueba de concepto es tratar la estructura de capital de la cartera como si se mantuviese fija contantemente. Esta simplificación sería equivalente a asumir que al final de cada periodo se desinvierte y vuelve a invertir (la cantidad desinvertida más el flujo positivo o negativo de ese determinado periodo) según la estructura de capital inicial. En la práctica, aunque también depende entre otras de la duración de los periodos, esto no sería practico, puesto que el coste de realizar operaciones de desinversión y reinversión constantemente podría minar los beneficios de las inversiones, volviendo una inversión rentable en una, si no irrentable, menos beneficiosa debido al coste de comisiones. Igual que en el caso anterior, se trata de decisiones personales que se deberían tratar en cada caso específico.

- **Solo largos.** Solo está permitido tomar posiciones largas en los activos, es decir, comprarlos. No se puede “apostar” a que el precio de estos descenderá para ganar ese margen de caída. Las posiciones cortas, es decir, vender un activo sin poseerlo con la esperanza de poder recomprarlo próximamente a un menor precio para devolver el subyacente, requieren garantías de capital diferentes que las hacen ser tratadas aparte. Al igual que en los casos anteriores, es una decisión personal que deberá ser tratado según cada caso, pero para simplificar el funcionamiento del algoritmo solo se permiten posiciones largas en la prueba de concepto.
- **Solo 10 activos financieros.** Como se comentó anteriormente, lo ideal para obtener los mejores resultados por parte de esta herramienta sería tener una base de datos muy completa, que dispusiese de diferentes activos financieros como deuda, acciones e índices de mercados muy variados. Sin embargo, este factor afecta notablemente al coste computacional del algoritmo, por esto mismo en este trabajo se ha decidido simplificar y usar tan solo 10 activos financieros. Como en casos anteriores, esto es fácilmente escalable y podría incrementarse el número de activos disponibles sin grandes complicaciones.

- **Sin tener en cuenta el coste de la operativa ni impuestos.** Los costes impuestos por el bróker y los impuestos son un lastre para la rentabilidad. En un escenario real sería muy importantes tenerlos en cuenta, sin embargo, no se tienen en cuenta este tipo de costes a la hora de desarrollar la herramienta. No obstante, no sería complicado añadirlos y en caso de profesionalizar la herramienta debería ser imprescindible ajustarla según el marco regulatorio de cada región de operación y según los costes del bróker con el que se opere.
- **Sin límite de compañías en cartera.** Por diversos motivos, como el coste de comisiones, es poco probable que un inversor quiera tener en su cartera un muy alto número de activos financieros. Sin embargo, incentivado por las dos asunciones comentadas anteriormente, el algoritmo no tendrá un límite de compañías en cartera. Igualmente es una decisión que debe ser tratada de manera personal en cada caso particular.
- **Periodos mensuales.** En la herramienta se trabaja con periodos mensuales, esto es que los flujos de caja se tienen en cuenta de manera mensual, asimismo las rentabilidades de la cartera se calculan también de manera mensual y, por lo tanto, como se comentó en un punto anterior, la reestructuración de la cartera para mantener la estructura fija inicial se hará también cada mes. Esta decisión como gran parte de las anteriores debe tratarse de manera personal, no obstante, no es una simplificación como tal, sino una asunción. Un umbral de un mes parece bastante coherente, pero dependiendo del caso específico podrían usarse periodos trimestrales, quincenales, etc. teniendo en cuenta que mientras menor sea el periodo mayor será el coste por la operativa.
- **Umbral de 20 meses.** Para la prueba de concepto se va a trabajar con casos en los que se dispone de una previsión de los flujos de caja mensuales para los siguientes 20 meses. Al igual que la anterior es una decisión que debe ser tratada en cada caso personal según los requerimientos del cliente.
- **Simulación de comportamiento de activos mediante un modelo estocástico.** Para el correcto funcionamiento de la herramienta es necesario realizar predicciones del comportamiento de los activos en el futuro. Estas predicciones se realizan en base a su comportamiento pasado, en la prueba se realizan mediante un modelo SDE (Stochastic Differential Equation). No obstante, en una versión

avanzada de la herramienta podría ser una opción realizar estas predicciones del comportamiento futuro de los activos mediante otras herramientas, por ejemplo, basadas en redes neuronales. Como se ha dicho esto no forma parte del objetivo del trabajo, pero será comentado en la sección de trabajo futuro.

Una vez descritas las asunciones y simplificaciones, el problema que se pretende resolver podría resumirse en: dados una cantidad de dinero disponible inicial y unos flujos de caja (20), ¿Qué combinación de los 10 activos financieros dados es más probable que pueda hacer frente a todos los flujos de caja negativos? Entendiendo hacer frente a un flujo de caja negativo que cuando esta cantidad es descontada del valor en ese momento de la cartera, el valor de esta siga siendo positivo.

## Modelado de la solución

La herramienta desarrollada se trata, como se ha comentado en apartados anteriores, de un algoritmo evolutivo. Los individuos que forman la población, es decir, las soluciones candidatas, son posibles composiciones de la cartera de inversión. En nuestro caso simplificado se tratará de un vector de 10 posiciones, en el cual cada posición representa a un activo. En cada posición del vector se almacenará un número real comprendido en el intervalo  $[0,1]$ , y cuya suma (la de todos los valores del vector) es exactamente igual a 1. Este número representa el peso del activo en cuestión sobre el total de la cartera, es decir, este número multiplicado por el capital total nos daría la fracción del capital a invertir en este activo.

La función de fitness o de adaptación es la que se utiliza para medir cuan bien una solución candidata se ajusta al medio, es decir, para cuantificar y poder distinguir una mejor solución de otra. Asimismo, la selección de individuos padre para generar las generaciones posteriores se basa también en esta función de adaptación. En la herramienta, la adaptabilidad al medio se mide según la probabilidad de hacer frente a todos los flujos de caja negativos. Para determinar el valor fitness de cada solución candidata se sigue un método Montecarlo. Este método nos da soluciones aproximadas mediante la simulación de pruebas aleatorias repetidamente. En otras palabras, se realizan

muchas simulaciones mediante un modelo SDE (Stochastic Differential Equation) con una componente aleatoria, como se ha comentado anteriormente. Entonces, enfrentaremos cada solución candidata a un determinado número de posibles escenarios (entendiendo escenario como la suma de las simulaciones de todos los activos); la función fitness será el porcentaje de esos escenarios en los que la solución candidata puede hacer frente a todos los flujos de caja negativos. A modo de ejemplo, supongamos que se simulan 100 escenarios posibles, es decir, 100 conjuntos de la simulación del comportamiento de los 10 activos seleccionados; si una solución candidata es capaz de hacer frente a todos los flujos de caja negativos en 65 de los escenarios simulados su función de fitness sería del 65%.

La selección de individuos padre se realiza mediante un mecanismo de selección por torneo. El tamaño del torneo y otros parámetros como la probabilidad de mutación que se comentarán más adelante, no están prefijado siendo posible regularlos en diferentes pruebas para encontrar la combinación que mejor resultado nos dé. No obstante, no es el objetivo de este trabajo encontrar la combinación optima, puesto que cada caso concreto deberá ser tratado específicamente y trabajar con sus datos y las pruebas que se realizan de la herramienta desarrollada buscan demostrar su funcionalidad más que optimizarla. Lo que, si se mantiene constante, forma parte del diseño del algoritmo es el elitismo. El mejor individuo de cada generación siempre pasará a la siguiente generación íntegro, sin ser cruzado ni mutado.

Se hace uso de diferentes operadores genéticos de reproducción. En primer lugar, se realiza un cruce entre dos individuos padres con una probabilidad prefijada en cada ejecución. El cruce que se realiza es un 1-point crossover, es decir un cruce en un punto. Se selecciona aleatoriamente una posición entre la segunda y la penúltima, en nuestro caso entre la posición 2 y la 9 (1 y 8 si empezamos a contar por el 0, en el lenguaje usado, Matlab, se hace a partir del 1). A continuación, la primera parte del vector padre, que contendrá la primera posición y una longitud entre de entre 1 y 9, se unirá con la segunda parte del vector madre, el cual contendrá su última posición y tendrá una longitud de 10 – longitud de la primera parte del vector padre, dando lugar a un vector hijo de longitud 10. Seguidamente se realizará el mismo proceso cambiando al padre y la madre, es decir, tomando la primera parte del vector madre y la segunda del vector padre. Para realizar esto el algoritmo procede de la siguiente manera: primero realiza un torneo y selecciona un individuo padre. Con una probabilidad  $p$  de cruzar definida antes de la ejecución

realiza otro torneo y selecciona entonces un individuo madre. A continuación, realiza un cruce, el vector resultante tendrá su primera mitad del padre y la segunda de la madre. Si la nueva población que se está generando aún no ha llegado a su tamaño máximo (también prefijado antes de la ejecución) teniendo en cuenta el nuevo individuo hijo, entonces se genera un segundo hijo que tendrá su primera mitad como la del individuo madre y la segunda como el padre. Con probabilidad  $1-p$  no se seleccionará un segundo individuo y por lo tanto no se realizará cruce, el hijo pasará a la siguiente generación idéntico al padre, por ahora, puesto que posteriormente se realiza una operación de mutación.

Posteriormente, sobre el o los hijos resultantes, se realizará una operación de mutación. Esta consiste en modificar cada entrada del vector hijo con una probabilidad  $q$  prefijada antes de la ejecución. En caso de mutación la entrada en cuestión es sustituida por un valor aleatorio entre 0 y 1.

Finalmente, el último operador utilizado es un tipo de mutación, una permutación. Con una probabilidad  $r$  una posición cualquiera del vector hijo será intercambiada con cualquier otra. Este operador tiene especial sentido en la herramienta debido a la naturaleza de esta, es decir, si no olvidamos que nuestro individuo es un vector en el cual cada posición representa a un activo financiero, parece una buena manera de mutar el intercambiar dentro de una cartera en la que el resto se mantiene constante el peso en ella de dos activos.

Como último paso una vez que la siguiente generación esta creada, hay que normalizar la cartera, es decir, hacer que la suma de todas las posiciones del vector sea 1. No hay que olvidar que nuestros individuos representan el 100% de una cartera de inversión, sería absurdo que la suma fuese distinto a 1. Esto implicaría apalancamiento, lo cual no se trata. La manera de reajustar los valores de la cartera para que su suma sea 1 es sencilla, se divide cada posición por la suma de todas ellas.

El proceso por lo tanto del algoritmo resultaría en:

1. Se inicializan todas las variables que van a determinar un caso. Se establecen unos flujos de caja, se cargan los datos históricos de los activos financieros que vayan a ser utilizados, se determina la condición de parada (número de generaciones que se van a producir), se determina el tamaño de la población, se determina el número de escenarios/simulaciones que se van a utilizar para evaluar el valor fitness de cada individuo en cada generación, se establece el tamaño de torneo y se

determinan las probabilidades de cruce, mutación y permutación definidas anteriormente.

2. Se genera una población inicial aleatoria del tamaño prefijado en el paso anterior. En otras palabras, se crea una matriz con tantas filas como sea el tamaño de la población y tantas columnas como históricos de activos financieros se hayan introducido. Se normaliza de manera que la suma todas las posiciones de cada fila sea 1.
3. Se evalúa el fitness de cada individuo generado. Esto se hace realizando tantas simulaciones de comportamientos de activos como hayan sido predeterminadas en el paso 1 y comprobando en cuantas de ellas los individuos podrían hacer frente a todos los flujos de caja negativos. Esto se hace multiplicando el valor de la cartera antes de un periodo por los incrementos de cada activo que forma la cartera en ese periodo ponderados según la estructura de la misma. Tras esto se le suma el valor del flujo de caja correspondiente al periodo, si el resultado es negativo equivaldría a que la empresa no podría hacer frente a los pagos, “quebraría” y así sucesivamente con todos los escenarios simulados para finalmente obtener un valor fitness de cada individuo.
4. Si se da la condición de parada, es decir, si esta generación era la prefijada en el paso uno como condición de parada, el algoritmo pararía y mostraría el individuo con mejor valor de adaptación, es decir, mostraría la composición de la cartera recomendada y su probabilidad de hacer frente a todos los flujos de caja en cualquier escenario. Si no se da la condición de parada se pasaría a la creación de la siguiente generación.
5. El mejor individuo de la generación actual pasaría directamente a la siguiente y a continuación se irían seleccionando por torneo los padres que, tras aplicar los operadores genéticos descritos anteriormente (cruce, mutación y permutación), irían pasando a formar la siguiente generación. Cuando la nueva población tuviese el tamaño determinado en el paso 1 se normalizaría y se volvería a ejecutar 3-4-5 hasta cumplirse la condición de parada.

## Recopilación de datos

Los 10 activos financieros seleccionados son 10 empresas españolas que han formado parte del IBEX35 desde su fundación hasta la actualidad. Estas son: Acerinox (ACX), Bankinter (BKT), BBVA (BBV), Endesa (ELE), Iberdrola (IBE), Mapfre (MAP), Repsol (REP), Santander (SAN), Telefónica (TEF) y Viscofan (VIS). Los motivos para seleccionar estas empresas han sido la facilidad para acceder a sus datos de cotizaciones históricas y la presunción de que empresas que han sido capaces de mantenerse en el índice de referencia español durante más de 25 años habrán tenido un comportamiento bursátil aceptable. No obstante, el hecho de seleccionar 10 empresas, aunque de diferentes sectores, del mismo mercado; y no haber dado ninguna opción de deuda ni de índices, puede provocar sesgos del tipo de que todas las empresas estén muy correlacionadas, pero este asunto se tratará en las conclusiones.

Debido a la naturaleza del problema los únicos datos necesarios son los históricos de los activos financieros utilizados, puesto que los flujos de caja serán seleccionados para realizar pruebas sin necesidad de representar un caso real. Su utilidad es la de poder simular posibles comportamientos futuros de estos en función de su comportamiento pasado. En un caso real de una ejecución serían necesarios también recibir por parte del usuario/cliente su estimación de flujos de caja, en la prueba de concepto realizada en este trabajo esos datos son elegidos por parte del desarrollador para contrastar el correcto funcionamiento del programa. Se hacen pruebas de varias configuraciones de flujos de caja diferentes para ver los diferentes comportamientos.

Se obtuvieron los históricos de cotización diaria de las 10 empresas con las que se trabaja a través de la página [invertedia.com](http://invertedia.com). Como se ha comentado en apartados anteriores, los periodos utilizados para realizar las pruebas en el trabajo son mensuales, por lo tanto, se adaptaron los datos recibidos para tan solo usar los cierres mensuales. Al disponerse de los datos de cotización desde la fundación del Ibex (enero del 92) se han usado todos ellos para realizar las simulaciones de escenarios futuros. Esto puede provocar algunos sesgos puesto que se podría decir que el comportamiento reciente afecta más que el pasado, pero dado que el objetivo del trabajo no es construir una cartera óptima para un caso real, sino demostrar la utilidad de la herramienta, puede usarse este tamaño del historial, no obstante, esto se comentará en las conclusiones.

## Capítulo 4. Estudio empírico

### Descripción del código y su estructura

El código ha sido desarrollado en el lenguaje Matlab. Se ha elegido este lenguaje por sus abundantes librerías y funciones predesarrolladas unido a la posibilidad de usarlo de forma gratuita gracias a la universidad.

Se ha fragmentado el código en varias funciones simples para poder facilitar su comprensión y testeo de errores a través de pruebas unitarias a cada función. Cada caso, entendiendo por caso unos determinados flujos de caja, activos financieros, etc. se codifica como un script el cual llama a la función *seleccionCartera* y esta realiza todo el proceso devolviendo la cartera optima construida con su valor fitness, es decir, la probabilidad de no quebrar de esa cartera. Este script comienza cargando los datos históricos de los activos financieros que vayan a ser usados en la ejecución, esta carga se realiza a través de un archivo Excel o csv previamente preparado con los datos necesarios. También en este script se introducen los flujos de caja del caso y todos los valores relativos a el algoritmo genético que se irán variando a lo largo de diferentes ejecuciones para encontrar los valores óptimos. Estos valores son: el número de generaciones, el tamaño de la población, el número de escenarios (cantidad de predicciones de los activos que se van a usar para medir el fitness), tamaño del torneo, probabilidad de cruce, probabilidad de mutación y probabilidad de permutación. El script muestra por pantalla todos los datos del caso, llama a la función *seleccionCartera* como se ha dicho anteriormente y posteriormente muestra por pantalla el resultado recibido.

La función *seleccionCartera* recibe todos los datos característicos del caso y genera la población inicial aleatoria, una matriz con tantas filas como el valor tamaño de población y haya recibido y tantas columnas como el número de activos. A continuación, inicia un bucle, el cual realizará tantas iteraciones como número de generaciones se hayan definido. En cada una de estas iteraciones se normalizará la población, se hará que la suma del valor en todas las posiciones de cada fila sea igual a uno, como se explicó en apartados anteriores. Seguidamente se llama a la función *simulación* la cual devuelve el valor fitness de cada individuo y será descrita más abajo. Con estos valores fitness de la población se



calcula y almacena la media y el máximo para, al final de la ejecución mostrarlos por pantalla y ver la evolución. A continuación, se llama a la función *reproducción*, función que será descrita más abajo y genera una nueva población en base a los valores fitness calculados anteriormente. Una vez se dispone de la nueva población se vuelve a iterar hasta que después del número prefijado de generaciones se sale del bucle, se muestra por pantalla la evolución que ha tenido el fitness medio y máximo de cada generación y se devuelve al script que la había invocado el individuo óptimo junto con su valor fitness.

La función *reproducción* que ha sido llamada por la función *seleccionCartera* es la encargada de, con unos individuos y valores de fitness dados, generar la siguiente generación. Para esto hace uso de la selección por torneo y operadores genéticos. La selección por torneo y los operadores genéticos de reproducción fueron desarrollados como diferentes funciones. *reproduccion* crea la matriz en la que almacenar la nueva generación. En primer lugar, añade el mejor individuo de la población anterior a esta debido al elitismo. A continuación, inicia un bucle que itera mientras la población no esté completa, en cada iteración selecciona a un padre usando la función *torneo* que lo seleccionara a través de un torneo. Seguidamente con una probabilidad de cruce seleccionará también por torneo un segundo padre y lo cruzará con el anterior para dar lugar a dos individuos hijos. Estos individuos hijos (o el padre si no se dio la probabilidad de cruce) serán a continuación mutados y permutados mediante las funciones *mutacion* y *permutacion* respectivamente cuyo funcionamiento ya ha sido explicado anteriormente y con probabilidades de mutación y permutación dadas. Cuando la nueva generación tiene un tamaño igual al determinado al inicio de la ejecución se la pasa a *seleccionCartera* para continuar la ejecución.

## Pruebas

En este apartado se van a mostrar algunos de los casos utilizados para probar el algoritmo y los resultados obtenidos con diferentes configuraciones de los parámetros. En el apartado siguiente se analizarán estos resultados.

El primer caso utilizado para probar el funcionamiento de la herramienta es el siguiente. Se utilizan los 10 activos financieros comentados anteriormente y los flujos de caja

utilizados son los que muestra la Figura 12. Se comienza con 1400 unidades monetarias en caja y se cancelando flujos positivos y negativos indistintamente hasta llegar a un flujo final de -1200 unidades monetarias. La suma de todos los flujos de caja del escenario, teniendo en cuenta la cantidad inicial o valor en el periodo 0, es de -200. Es un escenario bastante exigente puesto que el umbral de tiempo es de 20 meses. Conseguir un beneficio comenzando con 1400 u.m. de 200 en tan solo 20 meses es aproximable a una rentabilidad anual de más de un 8%, no es exactamente así debido a que continuamente están afectando los flujos de caja. En cualquier caso, ser capaz de mantener una rentabilidad de alrededor de un 8% (por usarlo de referencia), sistemáticamente sea cual sea el escenario con una cartera fija es muy difícil.

Se realizaron diferentes ejecuciones con distintas configuraciones de este caso. En la Figura 13 se ve la evolución del fitness (medio y máximo) según las generaciones en la ejecución con la siguiente configuración. A continuación, se copia el resultado mostrado por pantalla tras la ejecución (en pruebas posteriores se mostrará exclusivamente la información más relevante):

*Escenario compuesto por los 10 activos: ACERINOX, BANKINTER, BBVA, ENDESA, IBERDROLA, MAPFRE, REPSOL, SANTANDER, TELEFONICA y VISCOFAN.*

*Tamaño de la población: 100*

*Número de generaciones: 50*

*Número de previsiones estudiadas en cada generación: 7500*

*Tamaño de torneo: 5*

*Probabilidad de cruce: 0.500000*

*Probabilidad de mutación: 0.050000*

*Probabilidad de permutación: 0.200000*

*Flujos de caja previstos en el escenario:*

*Capital inicial:           1400*

<i>Periodo 1:</i>	<i>-100</i>
<i>Periodo 2:</i>	<i>-200</i>
<i>Periodo 3:</i>	<i>300</i>
<i>Periodo 4:</i>	<i>-500</i>
<i>Periodo 5:</i>	<i>200</i>
<i>Periodo 6:</i>	<i>600</i>
<i>Periodo 7:</i>	<i>-50</i>
<i>Periodo 8:</i>	<i>-150</i>
<i>Periodo 9:</i>	<i>-300</i>
<i>Periodo 10:</i>	<i>200</i>
<i>Periodo 11:</i>	<i>-800</i>
<i>Periodo 12:</i>	<i>300</i>
<i>Periodo 13:</i>	<i>200</i>
<i>Periodo 14:</i>	<i>-600</i>
<i>Periodo 15:</i>	<i>200</i>
<i>Periodo 16:</i>	<i>-100</i>
<i>Periodo 17:</i>	<i>200</i>
<i>Periodo 18:</i>	<i>-100</i>
<i>Periodo 19:</i>	<i>300</i>
<i>Periodo 20:</i>	<i>-1200</i>

#### *EVOLUCIÓN DE LA FUNCIÓN FITNESS EN LAS DIFERENTES GENERACIONES*

<i>#GENERACIÓN</i>	<i>FITNESSMEAN</i>	<i>FITNESSMAX</i>
<i>1</i>	<i>0.433579</i>	<i>0.470133</i>

2	0.450899	0.517600
3	0.466644	0.521733
4	0.476725	0.516133
5	0.491211	0.519867
6	0.492293	0.516400
7	0.501239	0.520000
8	0.508416	0.526267
9	0.496783	0.512533
10	0.506437	0.527200
11	0.511173	0.539067
12	0.518264	0.534000
13	0.520279	0.539600
14	0.516876	0.537333
15	0.514539	0.534267
16	0.524352	0.543867
17	0.513789	0.532000
18	0.515209	0.529600
19	0.523584	0.540000
20	0.523620	0.539333
21	0.515689	0.533733
22	0.515037	0.530533
23	0.526117	0.541067
24	0.511856	0.530000
25	0.521305	0.539200

26	0.521524	0.538000
27	0.521259	0.537333
28	0.511681	0.528533
29	0.527161	0.545600
30	0.527724	0.544267
31	0.504680	0.525200
32	0.509397	0.524800
33	0.520439	0.535467
34	0.514549	0.528533
35	0.519629	0.539333
36	0.520112	0.540533
37	0.506887	0.520533
38	0.516949	0.532267
39	0.526455	0.544400
40	0.511531	0.530400
41	0.519567	0.531200
42	0.504303	0.520800
43	0.514973	0.526000
44	0.519440	0.535067
45	0.529255	0.543467
46	0.521863	0.535733
47	0.515716	0.530533
48	0.509749	0.529200
49	0.512323	0.528133

50                      0.519280                      0.537333

*Cartera óptima resultante:*

*ACX: 4.720631%*

*BKT: 10.196559%*

*BBV: 0.444251%*

*ELE: 0.444522%*

*IBE: 9.322823%*

*MAP: 0.842130%*

*REP: 0.174016%*

*SAN: 0.178830%*

*TEF: 0.432722%*

*VIS: 73.243516%*

*Probabilidad de la cartera de hacer frente a todos los pagos previstos: 53.733333%*

Esta ejecución requirió de un tiempo de 2672s, unos 45 minutos y tan solo consta de 50 generaciones. En la Figura 13 se ve como inicialmente el valor fitness aumenta rápidamente, pero tras algunas generaciones pasa a oscilar entre el 0,53 y 0,54 sin mostrar una mejora clara.

En la Figura 14 se ve la gráfica de la evolución fitness durante las primeras 190 generaciones de la siguiente configuración:

*Tamaño de la población: 100*

*Número de generaciones: 1500*

*Número de previsiones estudiadas en cada generación: 5000*

*Tamaño de torneo: 15*

*Probabilidad de cruce: 0.800000*

*Probabilidad de mutación: 0.050000*

*Probabilidad de permutación: 0.500000*

*Cartera óptima resultante:*

*ACX: 4.353290%*

*BKT: 10.418478%*

*BBV: 0.075882%*

*ELE: 0.108109%*

*IBE: 9.096985%*

*MAP: 0.080967%*

*REP: 0.075927%*

*SAN: 0.071672%*

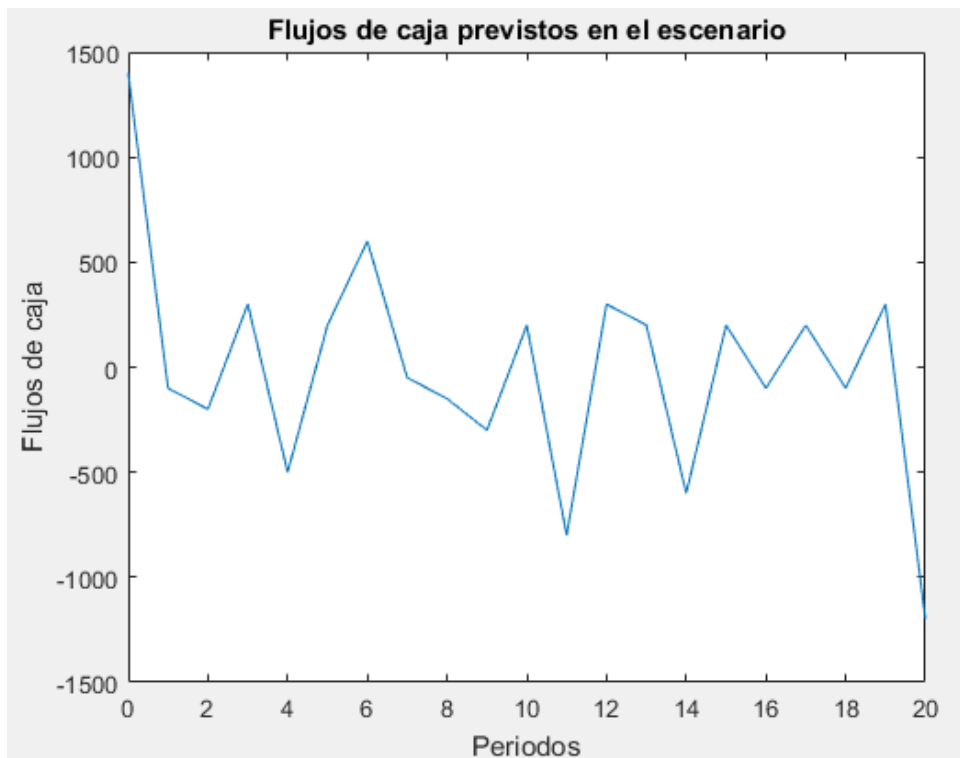
*TEF: 0.083741%*

*VIS: 75.634948%*

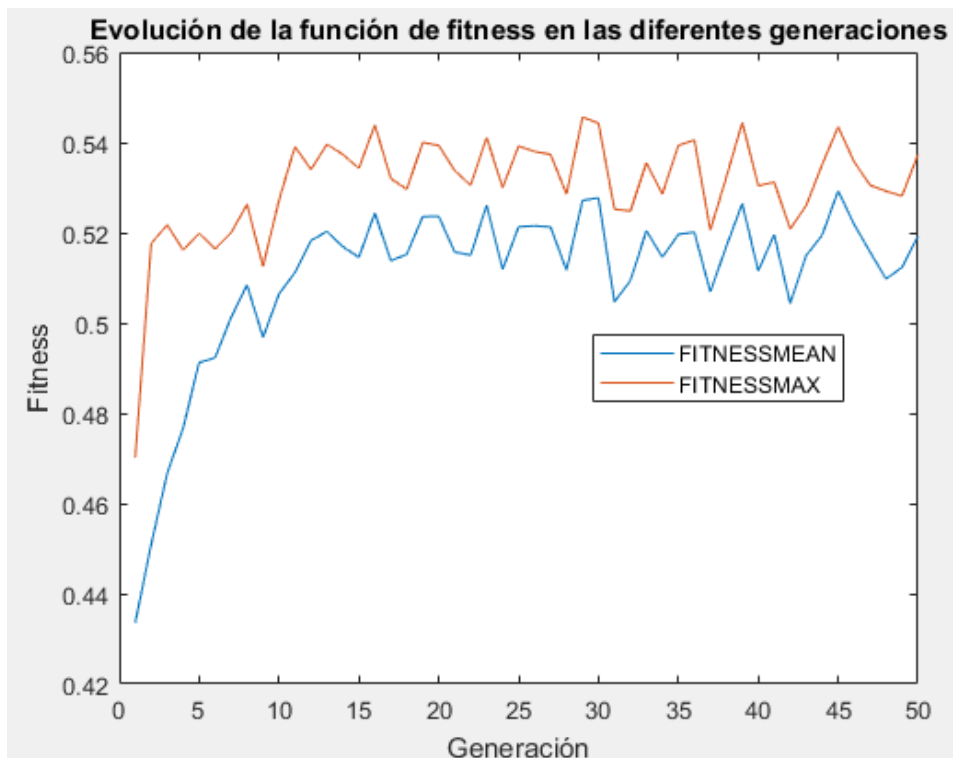
*Probabilidad de la cartera de hacer frente a todos los pagos previstos: 53.820000%*

En la Figura 14 solamente se han mostrado las 190 primeras generaciones puesto que la prueba realiza 1500 generaciones y la gráfica queda poco visual con más generaciones puesto que el valor del fitness continúa oscilando entre el 0,52 y 0,54.

Se ve que usando ambas configuraciones el resultado ha sido muy similar, pero a que la ejecución de la segunda fuese mucho más costosa.

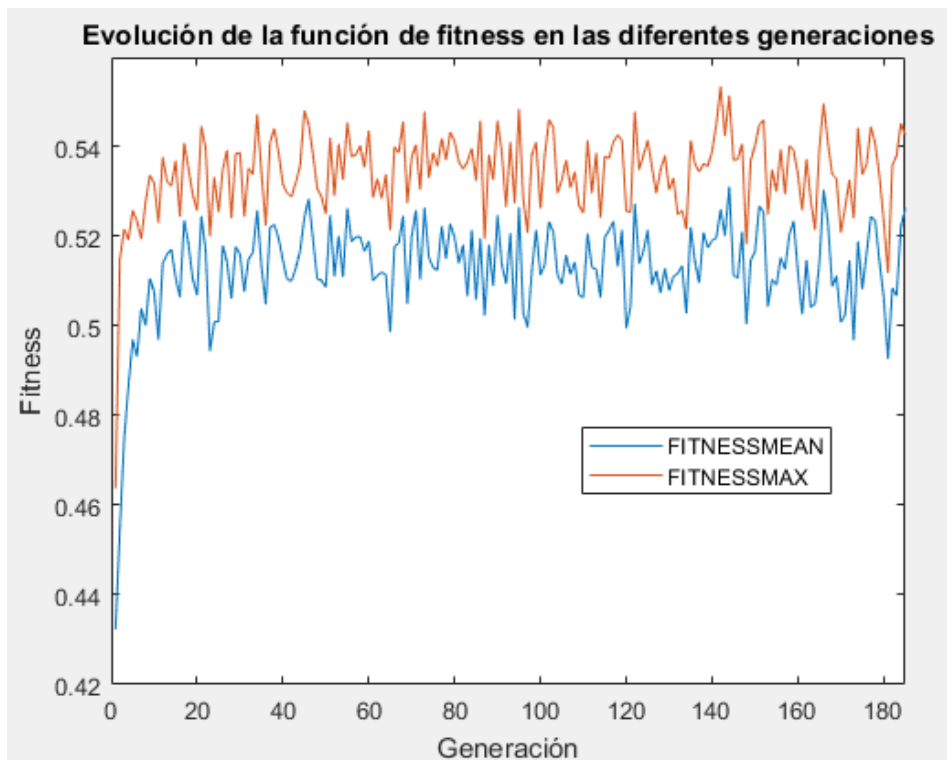


**Figura 12.** Gráfico de los flujos de caja en el caso de prueba 1.



**Figura 13.** Gráfico de la evolución del fitness en el caso de prueba 1.





**Figura 14.** Gráfico de la evolución del fitness en el caso de prueba 1.

En la figura 15 se ve los flujos de caja del segundo caso probado. En este caso se refleja ese uso alternativo de la herramienta aplicado a finanzas personales en lugar de a compañías. Se comienza con un capital inicial de 5.000 unidades monetarias y se realizan ingresos mensuales de 500 unidades monetarias para finalmente en el periodo 20 tener el objetivo de poder obtener 15.500 unidades monetarias. Este caso equivale a obtener unos intereses anuales equivalentes de aproximadamente un 5%. Igual que en el caso anterior, no es sencillo con una cartera prefijada y con un umbral de tiempo de tan solo 20 meses obtener estas rentabilidades equivalentes.

En la Figura 16 vemos la evolución del valor fitness de la siguiente configuración:

*Tamaño de la población: 60*

*Número de generaciones: 100*

*Número de previsiones estudiadas en cada generación: 10000*

*Tamaño de torneo: 6*

*Probabilidad de cruce: 0.800000*

*Probabilidad de mutación: 0.100000*

*Probabilidad de permutación: 0.500000*

*Flujos de caja previstos en el escenario:*

*Capital inicial: 5000*

*Periodo 1: 500*

*Periodo 2: 500*

*Periodo 3: 500*

*Periodo 4: 500*

*Periodo 5: 500*

*Periodo 6: 500*

*Periodo 7: 500*

*Periodo 8: 500*

*Periodo 9: 500*

*Periodo 10: 500*

*Periodo 11: 500*

*Periodo 12: 500*

*Periodo 13: 500*

*Periodo 14: 500*

*Periodo 15: 500*

*Periodo 16: 500*

*Periodo 17: 500*

*Periodo 18: 500*

*Periodo 19: 500*

*Periodo 20: -15500*

*Cartera óptima resultante:*

*ACX: 5.740920%*

*BKT: 7.398212%*

*BBV: 0.045369%*

*ELE: 0.047587%*

*IBE: 24.487640%*

*MAP: 0.017081%*

*REP: 0.039105%*

*SAN: 0.055218%*

*TEF: 0.047895%*

*VIS: 62.120972%*

*Probabilidad de la cartera de hacer frente a todos los pagos previstos: 63.840000%*

En la Figura 17 vemos la evolución del fitness de la siguiente configuración:

*Tamaño de la población: 100*

*Número de generaciones: 100*

*Número de previsiones estudiadas en cada generación: 10000*

*Tamaño de torneo: 5*

*Probabilidad de cruce: 0.600000*

*Probabilidad de mutación: 0.050000*

*Probabilidad de permutación: 0.100000*

*Cartera óptima resultante:*

*ACX: 4.154326%*

*BKT: 5.819537%*

*BBV: 0.053989%*

*ELE: 0.240099%*

*IBE: 35.518519%*

*MAP: 0.263036%*

*REP: 0.051111%*

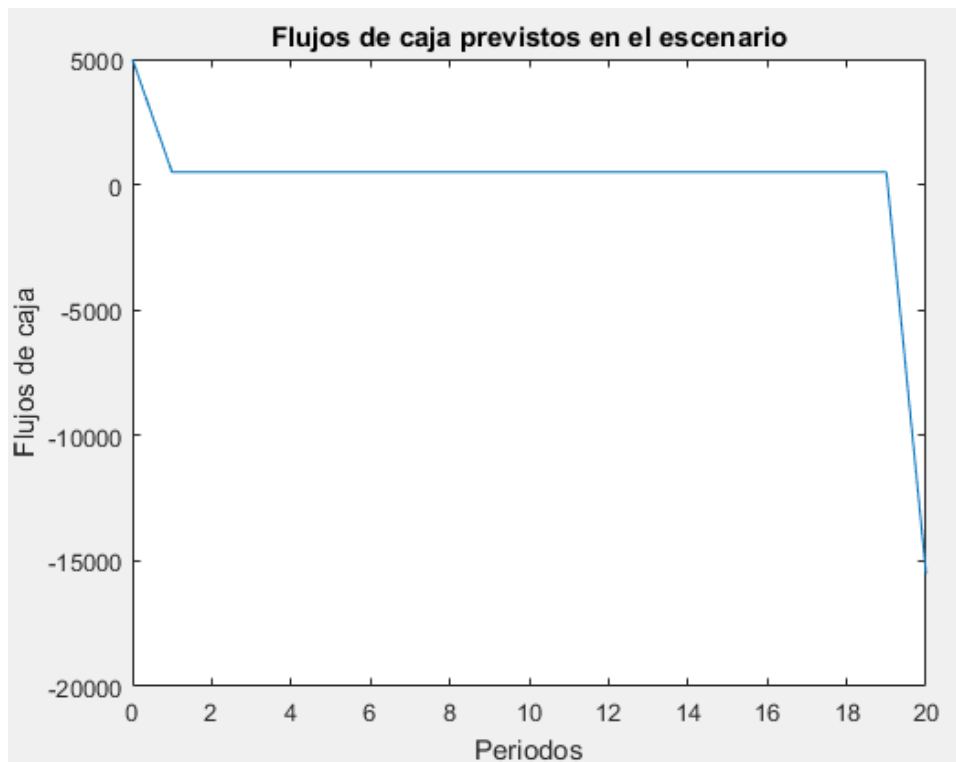
*SAN: 0.054473%*

*TEF: 0.280428%*

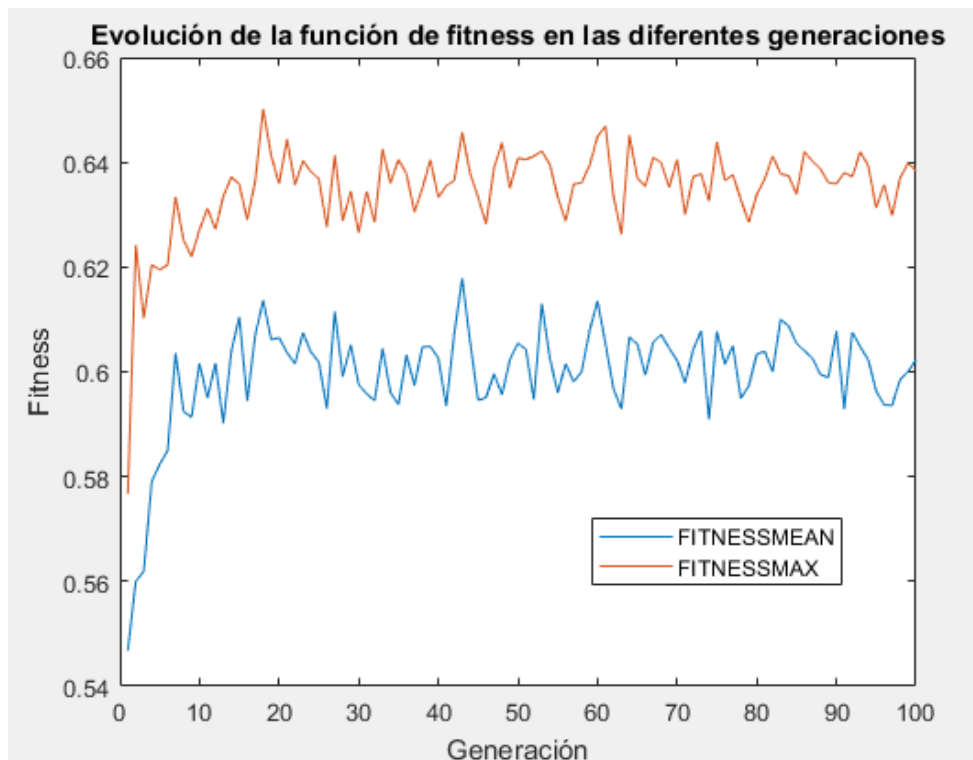
*VIS: 53.564482%*

*Probabilidad de la cartera de hacer frente a todos los pagos previstos: 64.350000%*

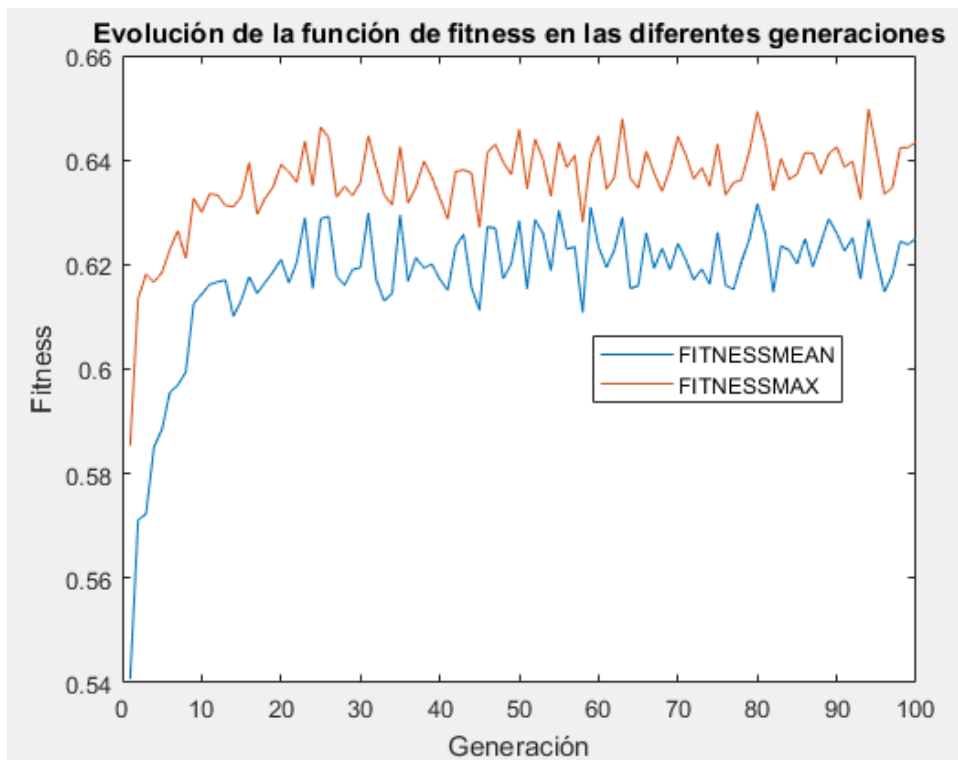
En ambos casos los resultados son bastante similares.



**Figura 15.** Gráfico de los flujos de caja en el caso de prueba 2.



**Figura 16.** Gráfico de la evolución del fitness en el caso de prueba 2.



**Figura 17.** Gráfico de la evolución del fitness en el caso de prueba 2.

En la Figura 18 se plantea un tercer en el que se comienza con un flujo de caja de 10.000 y se suceden bastantes más flujos de caja negativos que positivos, pero en el cómputo total la cantidad de flujos positivos teniendo en cuenta el capital inicial y negativos es de cero. Esto quiere decir que, manteniéndose fuera del mercado, es decir, sin comprar ningún activo financiero se podrían pagar todos los flujos de caja negativo. Sin embargo, como muestran las Figuras 19 correspondiente a la siguiente ejecución el algoritmo perdería dinero al invertir en un aproximadamente 28% de los escenarios.

*Tamaño de la población: 100*

*Número de generaciones: 50*

*Número de previsiones estudiadas en cada generación: 7500*

*Tamaño de torneo: 5*

*Probabilidad de cruce: 0.700000*

*Probabilidad de mutación: 0.010000*

*Probabilidad de permutación: 0.200000*

*Flujos de caja previstos en el escenario:*

*Capital inicial: 10000*

*Periodo 1: -500*

*Periodo 2: -1000*

*Periodo 3: 300*

*Periodo 4: -800*

*Periodo 5: 200*

*Periodo 6: 600*

*Periodo 7: -1200*

*Periodo 8: -700*

*Periodo 9: 300*

*Periodo 10: -2000*

*Periodo 11: -800*

*Periodo 12: -3000*

*Periodo 13: 300*

*Periodo 14: 600*

*Periodo 15: -2000*

*Periodo 16: 300*

*Periodo 17: -200*

*Periodo 18: 400*

*Periodo 19: -500*

*Periodo 20:*                   -300

*Cartera óptima resultante:*

*ACX:*   7.727004%

*BKT:*   9.293866%

*BBV:*   0.041253%

*ELE:*   0.044822%

*IBE:*   35.855044%

*MAP:*   0.046935%

*REP:*   0.592706%

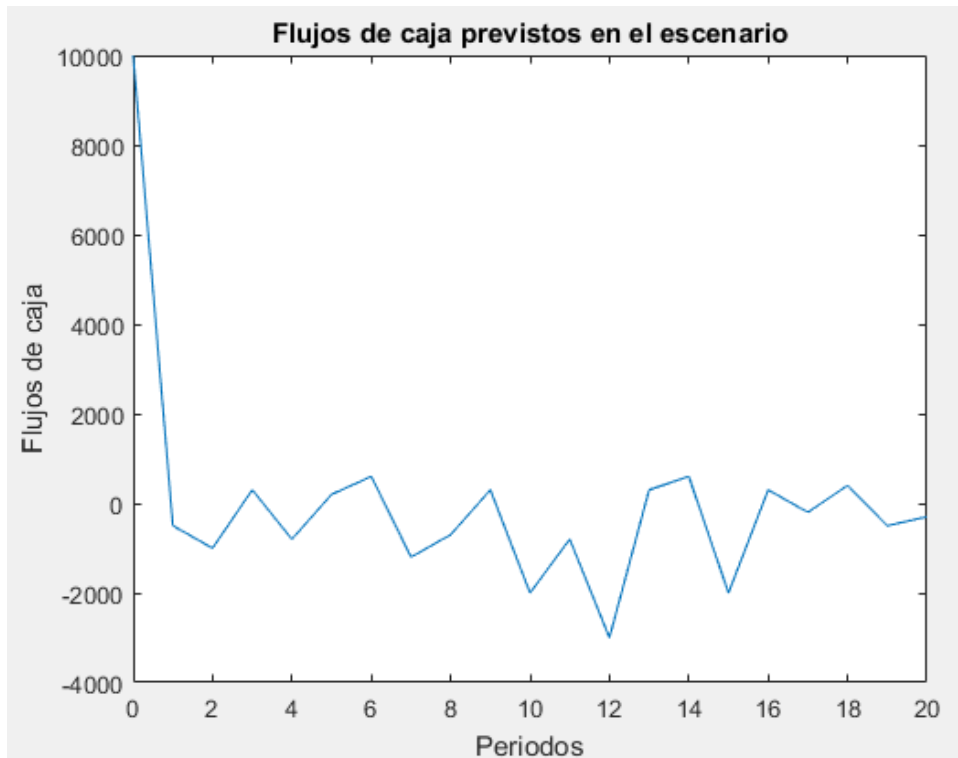
*SAN:*   0.045710%

*TEF:*   0.041732%

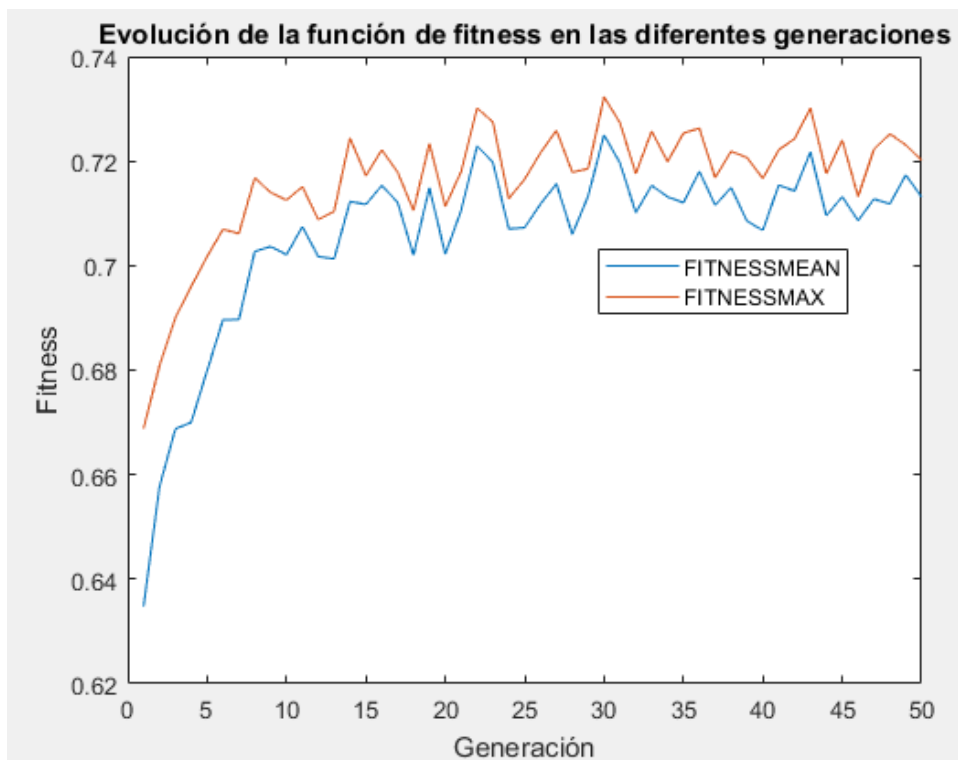
*VIS:*   46.310929%

*Probabilidad de la cartera de hacer frente a todos los pagos previstos: 72.013333%*





**Figura 18.** Gráfico de los flujos de caja en el caso de prueba 3.



**Figura 19.** Gráfico de la evolución del fitness en el caso de prueba 3.

## Evaluación de los resultados

El objetivo de las pruebas era demostrar el funcionamiento del algoritmo y en ningún lugar buscar la configuración óptima de los parámetros, esto se trataría de manera personalizada en cada caso real. Por lo tanto, podemos concluir que con las pruebas realizadas se ve el correcto funcionamiento del algoritmo. La prueba muestra como la inversión puede en bastantes escenarios evitar el impago y como el algoritmo optimiza ese resultado configurando una mejor cartera. No obstante, algunas características de los datos utilizados han marcado notablemente el resultado de las pruebas realizadas, introduciendo ciertos sesgos.

Lo primero que se observa en los resultados es que las carteras óptimas que se forman siempre tienen una forma similar. Están compuestas en entre un 50% y 75% por Viscofan y la parte restante por una combinación de Acerinox, Bankinter e Iberdrola, en la que Iberdrola suele estar más presente que las otras 2. Los otros 6 activos financieros suelen estar por debajo del 1%. Los motivos que llevan a esto son obviamente la elección de activos financieros realizada.

En primer lugar, tan solo se realizan las pruebas con acciones de 10 empresas española, esto provoca que haya grandes correlaciones entre ellas, es decir, que si la economía española va bien el valor de todas ellas aumente en bolsa y que baje en caso contrario. Esto provoca que en la simulación de escenarios la tendencia de todas sea la misma, impidiendo así invertir en activos que te cubran de la mala tendencia de otros porque en términos generales todos tienen la misma.

En segundo lugar, el historial usado para las simulaciones de todas las acciones comienza con la fundación del IBEX35 hace 25 años. Paralelamente, los datos utilizados no incluyen el pago de dividendos. La combinación de estos dos factores anteriores hace que algunas empresas, como los bancos se vean perjudicadas. Esto se debe a que los bancos (y otras empresas) dan suelen dar mucha importancia al pago de dividendos sistemático, lo que provoca que gran parte de sus beneficios sea retribuido directamente al accionista y no se vea reflejado en su cotización en bolsa. Si se hubiesen usado históricos con exclusivamente datos más recientes este efecto no habría sido marcado, pero al tratarse de 25 años afecta significativamente a la rentabilidad mostrada en bolsa puesto que solo mide la cotización sin tener en cuenta el pago de dividendos. Esto hace que Viscofan sea

con mucha diferencia la empresa más rentable de estos 25 años y por lo tanto el algoritmo construya carteras en las que se le da tantísimo peso.

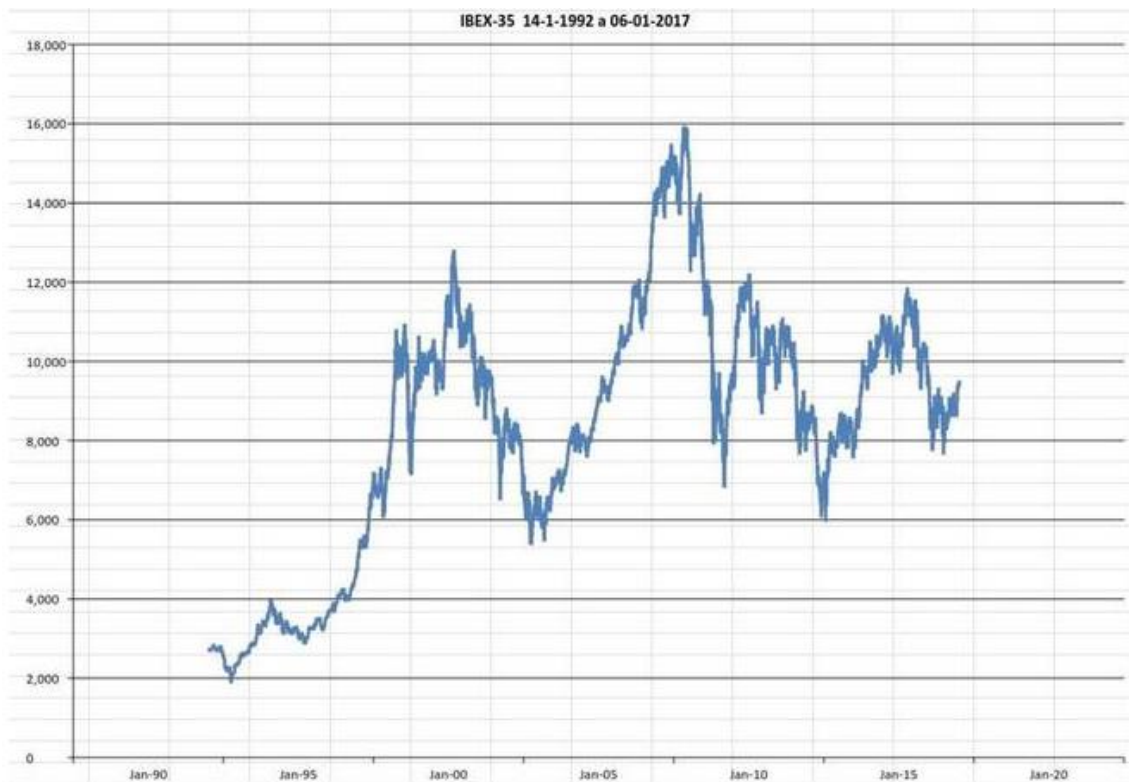
En tercer lugar, el diseño del algoritmo lleva naturalmente a la diversificación, es decir, a no concentrar mucho capital en un solo activo. Esto que a priori parece contradictorio con lo anterior, en realidad ha provocado que la presencia de Viscofan no sea aún mayor, como podría haber sido si el algoritmo estuviese diseñado de otra manera puesto que la rentabilidad de Viscofan es con mucha diferencia la mayor de las 10 empresas. Cuando se genera una población el último paso que se realiza en la normalización, es decir, que la suma de todas las posiciones del vector sea uno. Esto se realiza dividiendo el valor en cada posición por la suma de todas estas. Cuando se realiza un cruce o una permutación los valores cambian de posición, pero no se generan valores nuevos, con una mutación el valor máximo que puede darse en una posición es un 1 pero al normalizar posteriormente este va a verse muy reducido. Esto explica que en las carteras veamos varios activos con valores 0, los 0 son fáciles de mantener, pero que sea muy difícil que uno, Viscofan, pase del 75%. Esto puede parecer una característica negativa, pero tan solo debido al caso simplificado que estamos tratando. En una ejecución real concentrar mucho capital en un único activo conlleva mucho riesgo.

En las gráficas que muestran el fitness se ve una cierta volatilidad cuando converge en torno a unos valores, volatilidad en la que se ve una clara correlación entre el fitness máximo y la media. Esto puede ser explicado porque, aunque e hagan muchas simulaciones en cada generación (típicamente 7500 o 10000), pueden concentrarse mayores simulaciones negativas o positivas. Como se ha comentado anteriormente, todos los activos están muy correlacionados, por lo tanto, la cartera resultante funciona especialmente bien en los buenos escenarios, pero en los malos, al caer sistemáticamente todos los activos y no poder estructurarse de una manera que se proteja con unos de la caída de otros cae su fitness, tanto el máximo como el medio.

Otro aspecto destacable es que pese a haber realizado pruebas con un número variable de generaciones (se han realizado bastantes más de las comentadas en el apartado anterior), el valor de fitness tiende a estabilizarse y oscilar en un intervalo reducido antes de la generación 50. Esto parece provocado otra vez por las características de los datos usados. Sistemáticamente en las simulaciones de escenarios realizadas Viscofan tiene un comportamiento mucho mejor que el resto de los activos, esto hace que el algoritmo se centre maximizar la presencia de Viscofan y al no encontrar activos con los que cubrirse

de las caídas de Viscofan introduzca los siguientes más rentables. Una vez que la cartera está formada con un alto peso en Viscofan y el capital restante distribuido entre Iberdrola, Acerinox y Bankinter apenas se modifica porque apenas habría cambios en su valor fitness una vez que se da esta estructura. Además de esto, al solo disponer de 10 activos, de los cuales algunos están correlacionados con otros de mayor rentabilidad y por lo tanto nunca formarán parte de la cartera, no existe un dominio de combinaciones significativamente diferentes entre ellas muy amplio. En definitiva, que tan solo haya 10 activos y que estén bastante correlacionados entre ellos es lo que lleva a esta convergencia tan rápida.

Otro hecho a comentar es que en el Caso 3, en el cual la suma de todos los flujos de caja incluyendo el capital inicial es cero tan solo se logre un valor fitness de alrededor al 72%. En este caso se podría hacer frente a todos los pagos simplemente manteniéndose fuera del mercado puesto que se tiene capital suficiente, a diferencia de los otros dos casos en los que mantenerse completamente fuera del mercado equivaldría a realizar impago necesariamente. Sin embargo, al aplicar la limitación de invertir todo el capital disponible vemos que en un 28% de los escenarios simulados se va a perder dinero provocando el impago. Esto puede estar provocado por el corto umbral de las previsiones, es decir, que tan solo se mida a 20 meses unido con el histórico de datos utilizados. El Ibex desde su fundación se ha revalorizado un 255%, lo que equivale a un 6,82% anual. Sin embargo, este crecimiento no ha sido constante como se puede ver en la Figura 20. Desde su fundación ha pasado por dos grandes burbujas, la de las empresas tecnológicas alrededor del 2000 y la más reciente alrededor de 2007, las rentabilidades anuales en los años 2002 y 2008 fueron respectivamente de -28,11% y -39,43% [36]. Esto manifiesta que a la hora de realizar simulaciones puede ocurrir que para un periodo de tan solo 20 meses usando estos históricos se simulen comportamientos de rentabilidades negativas.



**Figura 20.** Evolución histórica del IBEX-35. Extraído de BME (Bolsas y Mercados Españoles).

## Capítulo 5. Planificación, Presupuesto y Marco regulador

### Planificación

Este apartado tiene como propósito detallar la planificación utilizada durante todo el desarrollo del proyecto. Se hace uso de un diagrama de Gantt para detallar la estimación del tiempo en cada fase del proyecto.

Inicialmente se fragmento el proyecto en una serie de etapas. La estimación temporal se ha realizado imputando a cada etapa del desarrollo una cantidad de horas estimadas. La carga de horas a cada tarea se hará por días de trabajo, con una equivalencia de un día igual a 6 horas. La elaboración del trabajo comienza el 12 de julio de 2017 y finaliza el 26 de septiembre del mismo año, equivalente a 55 días de trabajo, 330 horas.

Las etapas en las que se divide el proyecto son las siguientes:

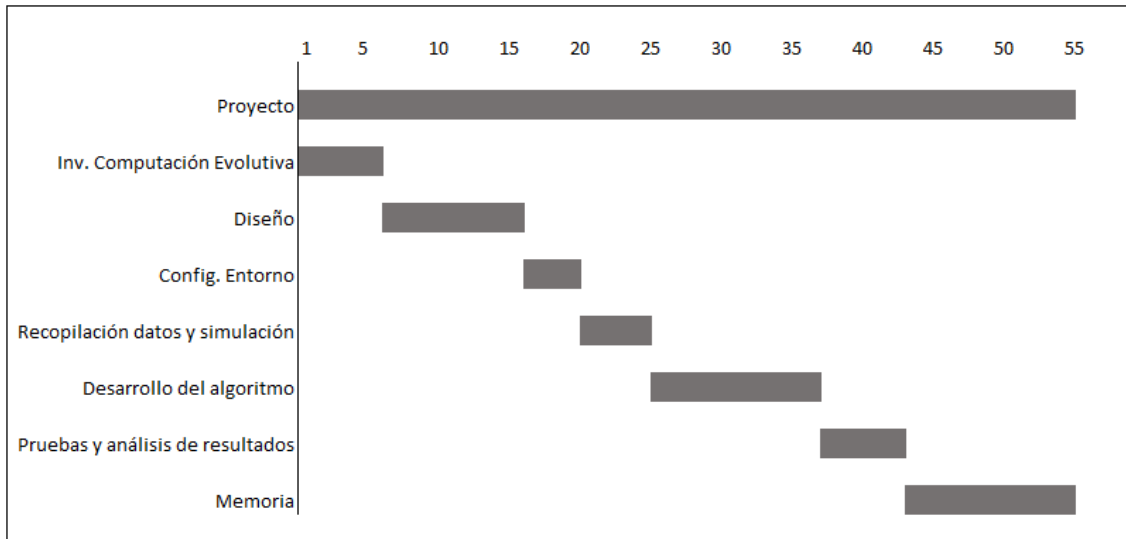
- Investigación sobre computación evolutiva: Esta fase incluye el proceso de investigación y aprendizaje a nivel teórico de los conceptos que se van a aplicar en fases futuras. Tiempo estimado: 6 días.
- Diseño de la herramienta: En esta fase se realiza el diseño de la herramienta marcando los objetivos y requisitos de la misma. Se realiza una comprensión del problema y se da un enfoque algorítmico para abordar la solución. Se realiza también una revisión de la literatura relacionada con la aplicación de la computación evolutiva en las finanzas. Tiempo estimado: 10 días.
- Configuración y familiarización con el entorno: En esta etapa se configura el entorno de trabajo Matlab para su uso en las siguientes fases. También se realizan algunos ejercicios y consultas al MathWorks para conocer mejor el lenguaje. Tiempo estimado: 4 días.
- Recopilación de datos y simulación de activos: Esta fase incluye la selección y recopilación de los datos a usar y de una primera fase de desarrollo de código en

la que se trabaja la generación de escenarios haciendo uso de estos datos. Tiempo estimado: 5 días.

- Desarrollo y testeo del algoritmo genético: En esta fase se codifica la solución planteada y se le realizan pruebas unitarias a este código. Tiempo estimado: 12 días.
- Fase de pruebas y análisis de los resultados: En esta etapa se realizan pruebas a la herramienta completa variando los parámetros para comprobar cómo afectan a los resultados las diferentes configuraciones del algoritmo. Tiempo estimado: 6 días.
- Elaboración de la memoria: Elaboración de un documento que agregue todo el trabajo realizado. Tiempo estimado: 12 días.

	Fecha Inicio	Fecha Fin	Duración	Espera
Proyecto	12/07/2017	26/09/2017	55	0
Inv. Computación Evolutiva	12/07/2017	19/07/2017	6	0
Diseño	20/07/2017	02/08/2017	10	6
Config. Entorno	03/08/2017	08/08/2017	4	16
Recopilación datos y simulación	09/08/2017	15/08/2017	5	20
Desarrollo del algoritmo	16/08/2017	31/08/2017	12	25
Pruebas y análisis de resultados	01/09/2017	08/08/2017	6	37
Memoria	11/09/2017	26/09/2017	12	43

**Figura 21.** Etapas del proyecto y duración estimada.



**Figura 22.** Planificación GANTT del proyecto.

## Presupuesto

Con la planificación temporal finalizada comienza la estimación del presupuesto para el mismo. El presupuesto es una parte fundamental porque va a determinar si un proyecto es o no rentable y por lo tanto si se va a llevar o no a cabo. En este apartado se detallan los costes en los que se estiman para el proyecto. Se realiza la estimación separando por personal y costes materiales. A nivel de personal intervienen en el desarrollo del proyecto Alejandro Dueñas Paz y David Quintana Montero. A nivel de coste material, el trabajo se desarrolla en un Lenovo Ideapad 100-15IBD usando el sistema operativo Windows 10 proporcionado gratuitamente por la Universidad Carlos III de Madrid a los estudiantes, el paquete de Microsoft Office 2016 también proporcionada por la universidad y Matlab, también proporcionado de manera gratuita por la universidad. Por lo tanto, a nivel de coste material tan solo se incurre en el ordenador puesto que todo el software esta proporcionado de manera gratuita por la UC3M.



<b>Coste personal</b>				
<b>Nombre</b>	<b>Puesto</b>	<b>Tarifa (€/h)</b>	<b>Horas</b>	<b>Coste total</b>
Alejandro Dueñas Paz	Ivestigador	25	330	8.250 €
David Quintana Montero	Asesor externo	500	10	5.000 €
<b>TOTAL</b>				<b>13.250 €</b>
<b>Coste equipo</b>				
<b>Descripción</b>	<b>Coste (€)</b>	<b>Tiempo en uso</b>	<b>Periodo amortización</b>	<b>Coste imputable</b>
Lenovo ideapad		426 3 meses	48 meses	26,63 €
<b>TOTAL</b>				<b>26,63 €</b>
<b>Concepto</b>				<b>Coste agregado</b>
Total sin impuestos				13.276,63 €
Impuestos (21%)				2.788,09 €
<b>TOTAL</b>				<b>16.064,72 €</b>

**Figura 23.** Presupuesto total del proyecto.

## Marco regulador

Para llevar a cabo la explotación de la herramienta sería imprescindible tener en cuenta la legislación vigente respecto a la inversión y la disponibilidad de los datos para elaborar un sistema robusto. Los datos que se usan en todo caso son históricos de las cotizaciones de productos financieros, datos que deben ser públicos. Se debería de tratar la legislación en estos aspectos en todos los países en los que se vaya a participar en el mercado. La normativa española relativa a los mercados de valores e instrumentos financieros se encuentra profundamente marcada por normas de carácter europeo tras la transposición de la Directiva de instrumentos financieros, conocida como MiFID. La transposición se realizó a través de dos normas: la Ley 47/2007 que supuso una importante modificación de la Ley 28/1988, del Mercado de Valores español (LMV) y, el Real Decreto 217/2008, que estableció el régimen jurídico aplicable a las entidades que prestan servicios de inversión [37].

En cuanto a la inversión en bolsa, en España la potestad normativa se puede clasificar en:

- La legislación común, (código penal, civil, mercantil) la legislación común, (Código penal, civil, mercantil) Ley de Sociedades Anónimas y demás reglas que consideran los derechos e intereses de los accionistas y obligacionistas, y regulan sus relaciones internas.

- La legislación especial, constituida por la Ley 24/88 del 28 de Julio sobre Mercado de Valores y disposiciones reglamentarias que la desarrollan.

## Capítulo 6. Conclusiones

### Conclusiones

El trabajo partía con dos objetivos primordiales. Por un lado, realizar un análisis de la aplicación de la computación evolutiva en las finanzas, esto fue desarrollado en el Capítulo 2 mostrando que se hace uso de estas técnicas en diferentes áreas como la predicción financiera, evaluación del riesgo de crédito, optimización de carteras, valoración de activos o trading algorítmico con resultados muy positivos viendo el incremento que se está dando.

Por otro lado, el verdadero grueso del trabajo recaía en el desarrollo de una herramienta propia que aplicase la computación evolutiva a la gestión de riesgos financieros. La herramienta, con todas las limitaciones y simplificaciones asumidas, ha sido probada con diferentes casos y funciona correctamente. Por lo tanto, podría concluirse que el trabajo ha cumplido con sus objetivos satisfactoriamente.

Como se comentó en el apartado de evaluación de los resultados, estos se han visto un poco sesgados por las limitaciones implementadas. No obstante, cumplen su objetivo planteado con la descripción del problema. Se planteaban unos flujos de caja que, salvo en uno de los casos, implican el no poder cumplir con los flujos de caja negativos. El algoritmo ha demostrado encontrar carteras de inversión que salvarían esta situación y mejorarlas a lo largo de generaciones hasta encontrar un valor óptimo.

### Trabajo futuro

Como se ha visto en el apartado anterior la herramienta es funcional, por lo tanto, como prueba de concepto la conclusión es satisfactoria. Sin embargo, de cara a usar la herramienta en un entorno profesional habría que tener en cuenta ciertos factores y trabajar en ellos. Estos son los que se va a comentar en este apartado.

1. En primer lugar, la oferta de activos financieros tiene que aumentar. En lugar de trabajar con acciones de compañías directamente probablemente fuese más eficiente trabajar con índices. Ya sean estos índices geográficos, al menos debería darse la opción de invertir en índices estadounidenses, europeos y asiáticos, o índices por sectores. Por otro lado, sería necesaria también los productos de deuda, tanto pública como privada y con varias áreas geográficas disponibles.
2. Otra característica que incorporar sería el rebalanceo de la cartera en el tiempo. Es más complicado hacer frente a todos los flujos de caja negativos con una cartera fijada de manera estática en el instante 0, especialmente mientras mayor sea el umbral de tiempo previsto en los flujos de caja, que con una actualización constante de esta.
3. Podría probarse el realizar las simulaciones del comportamiento de los activos con modelos más elaborados, que hagan uso de la inteligencia artificial. Por ejemplo, si usamos redes de neuronas para la predicción del comportamiento de activos y estas demuestran predecir mejor los resultados obtenidos serían más preciosos. No hay que olvidar que los resultados obtenidos están basados en una simulación de muchos escenarios, cuanto más precisa sea esta simulación mejor serán los resultados.
4. En esta prueba no se ha dedicado demasiado esfuerzo a optimizar los parámetros para obtener así mejores resultados, en parte porque las características de los activos e históricos utilizados llevaban a que algoritmo convergiera rápidamente. En casos reales habría que tratar de manera personalizada y con profundidad la optimización de parámetros, entendiendo por parámetros las características del algoritmo genético, pero también el histórico utilizado puesto que su profundidad afecta mucho a las simulaciones.
5. Por supuesto eliminar las simplificaciones asumidas para la prueba de concepto. Como permitir largos y cortos siempre que el cliente este de acuerdo o permitir mantener parte del capital fuera del mercado.

## Impacto en el contexto socioeconómico

El contexto socioeconómico planteado en el Capítulo 1, con el auge de las fintech, supone un entorno ideal para el desarrollo de la herramienta. Una empresa en disposición de esta

herramienta podría dar servicio de asesoramiento financiero a pymes, ayudando en la gestión de sus riesgos financieros y servir como apoyo en la toma de decisiones en empresas de un mayor tamaño.

Paralelamente una segunda vía de negocio podría ser la de la planificación financiera personal. Muchos inversores buscan una gestión pasiva de su patrimonio, es decir, replicar un índice o un determinado benchmark. Destaca en esto Vanguard por ejemplo. Pero están apareciendo nuevas herramientas como Indexacapital que construyen una cartera personalizada en función del perfil de capacidad de asumir riesgo y aversión a este del cliente. La herramienta desarrollada podría usarse de manera similar al negocio de Indexacapital, pero con un enfoque diferente, en lugar de intentar maximizar el beneficio con un determinado nivel de riesgo, el enfoque de la herramienta planteada en este trabajo sería cumplir un objetivo en un determinado umbral de tiempo. Es decir, minimizar el riesgo con el que se consigue un determinado objetivo de capital. Probablemente este sistema de inversión fuese bien recibido puesto que la gente tiende a ser adversa al riesgo y va a sentirse más cómoda con un sistema que a menor riesgo le garantice un objetivo de rentabilidad que con otro que se pueda obtener mayor rentabilidad, pero pueda no alcanzar ese objetivo o incluso sufrir pérdidas.

La monetización de la herramienta podría realizarse de diferentes maneras. Por un lado, el negocio de asesoría financiera a empresas y apoyo en la gestión del riesgo podría desempeñarse mediante contratos de asesoría en un determinado marco temporal con un precio mensual determinado en función del volumen de trabajo. Por otro lado, la monetización del negocio del apoyo a la planificación financiera personal podría realizarse mediante el cobro de una comisión sobre los beneficios obtenidos, comisión que obviamente se tenga en cuenta para realizar el cálculo de la cartera óptima.

## Bibliografía

- [1] Inteligencia Artificial, el futuro del crecimiento. Accenture. Encontrado el 25 de septiembre de 2017 en [https://www.accenture.com/t00010101T000000Z\\_w\\_/es-es/acnmedia/PDF-16/Accenture\\_Inteligencia\\_artificial\\_el-futuro-del\\_crecimiento\\_esp.pdf](https://www.accenture.com/t00010101T000000Z_w_/es-es/acnmedia/PDF-16/Accenture_Inteligencia_artificial_el-futuro-del_crecimiento_esp.pdf)
- [2] Jiménez, M. D. J. P., & Caparrini, F. S. (2003). *Máquinas moleculares basadas en ADN* (No. 2). Universidad de Sevilla.
- [3] Brabazon, A., Michael, O., & Dempsey, I. (2008). An introduction to evolutionary computation in finance. *IEEE Computational Intelligence Magazine*, 3(4).
- [4] J.R. Koza, D. Andre, F.H. Bennett III, M. Keane, *Genetic Programming 3: Darwinian Invention and Problem Solving*. Morgan Kaufmann, 1999.
- [5] A.M. Turing, "Intelligent Machines," In Ince, D.C. (Ed.), 1992, *Mechanical Intelligence: Collected Works of A.M. Turing*, North-Holland, 1948, pp. 107–128.
- [6] J. Holland, *Adaptation in Natural and Artificial Systems*. Ann Arbor: The University of Michigan Press, 1975.
- [7] L. Fogel A. Owens, and M. Walsh, *Artificial Intelligence through Simulated Evolution*. New York: Wiley, 1966.
- [8] I. Rechenberg, *Evolutionsstrategie: Optimierung technischer Systeme nach Prinzipien der biologischen Evolution*. Stuttgart: Frommann-Holzboog, 1973.
- [9] H.-P. Schwefel, *Evolutionstrategie und numerische Optimierung*. Dissertation, Technische University at, Berlin, 1975.
- [10] D.E. Goldberg, *Genetic Algorithms in Search, Optimization and Machine Learning*. Reading, MA: Addison-Wesley, 1989.
- [11] K.A. De Jong, *Evolutionary Computation: A Unified Approach*. MIT Press, 2006.
- [12] J.R. Koza, *Genetic Programming*. MIT Press, 1992.

- [13] C. Darwin, *On the Origin of the Species by Means of Natural Selection, or the Preservation of Favoured Races in the Struggle for Life* (reprinted 1985). London: Penguin Books, 1859.
- [14] H. Spencer, *The Principles of Biology*. vol. 1, London and Edinburgh: Williams and Norgate, 1964.
- [15] Evolving Objects (EO): an Evolutionary Computation Framework. Encontrado el 25 de septiembre de 2017 en <http://eodev.sourceforge.net/>
- [16] How the Genetic Algorithm Works. MathWorks. Encontrado el 25 de septiembre de 2017 en <https://es.mathworks.com/help/gads/how-the-genetic-algorithm-works.html>
- [17] Rudolph, G. (1994). Convergence analysis of canonical genetic algorithms. *IEEE transactions on neural networks*, 5(1), 96-101.
- [18] Eiben, A. E., & Smith, J. E. (2003). *Introduction to evolutionary computing* (Vol. 53). Heidelberg: springer.
- [19] Santana Quintero, L. V., & Coello Coello, C. A. (2006). Una introducción a la computación evolutiva y alguna de sus aplicaciones en Economía y Finanzas. *Revista de Métodos Cuantitativos para la Economía y la Empresa*, 2.
- [20] Mitchell, M. (1998). *An introduction to genetic algorithms*. MIT press.
- [21] Goldberg, D. E. (1989). Genetic algorithms in search, optimization, and machine learning, 1989. *Reading: Addison-Wesley*.
- [22] Koza, J. R. (1992). *Genetic programming: on the programming of computers by means of natural selection* (Vol. 1). MIT press.
- [23] Algoritmos genéticos. Fernando Berzal. Encontrado el 25 de septiembre de 2017 en <http://elvex.ugr.es/decsai/iaio/slides/G2%20Genetic%20Algorithms.pdf>
- [24] Eiben, A. E., & Smith, J. E. (2003). *Introduction to evolutionary computing* (Vol. 53). Heidelberg: springer.
- [25] Auger, A. (2005). *Theoretical Computer Science*.
- [26] Jägersküpper, J. (2006). *Theoretical Computer Science*.

- [27] Hansen, N., & Kern, S. (2004, September). Evaluating the CMA evolution strategy on multimodal test functions.
- [28] Algos 3.0, *Developments in Algorithmic Trading*, *Traders Magazine* 2007. Special Report. SourceMedia's Custom Publishing Group.
- [29] R. Bauer, *Genetic Algorithms and Investment Strategies*. New York: Wiley, 1994.
- [30] A. Brabazon and M. O'Neill (eds), *Natural Computing in Computational Finance*. Springer: Berlin, 2008.
- [31] A. Brabazon, and M. O'Neill, (eds), *Computational Intelligence in Finance*. Springer: Berlin (forthcoming). 2009.
- [32] S.-H. Chen, *Genetic Algorithms and Genetic Programming in Computational Finance*. (ed)., Kluwer Academic Publishers, 2002.
- [33] S.-H. Chen, *Evolutionary Computation in Economics and Finance*. Physica-Verlag, 2002.
- [34] S.-H. Chen, P. Wang, and T.-W. Kuo, (eds) *Computational Intelligence in Economics and Finance (vol. II)*, Springer: Berlin, 2007.
- [35] Balance sheet management benchmark survey. PwC. Encontrado el 25 de septiembre de 2017 en <https://www.pwc.com/gx/en/banking-capital-markets/assets/balance-sheet-management-benchmark-survey.pdf>
- [36] El Ibex cumple 25 años: así ha evolucionado el principal selectivo de la bolsa española (El Confidencial). Encontrado el 25 de septiembre de 2017 en [https://www.elconfidencial.com/mercados/2017-01-13/el-ibex-cumple-25-anos-asi-ha-evolucionado-el-principal-selectivo-de-la-bolsa-espanola\\_1316005/](https://www.elconfidencial.com/mercados/2017-01-13/el-ibex-cumple-25-anos-asi-ha-evolucionado-el-principal-selectivo-de-la-bolsa-espanola_1316005/)
- [37] La regulación del mercado de valores español. La normativa europea MIFID. Encontrado el 25 de septiembre de 2017 en <http://www.bolsamadrid.es/esp/Inversores/MercadoEsp2.aspx>



## Glosario

**Acción:** Parte alícuota en que se divide el capital social de una empresa, que proporciona a su propietario una renta variable, dependiendo de los resultados obtenidos por la empresa.

**Activo financiero:** Título o simplemente una anotación contable, por el que el comprador del título adquiere el derecho a recibir un ingreso futuro de parte del vendedor.

**Apalancamiento financiero:** El apalancamiento financiero consiste en utilizar algún mecanismo (como deuda) para aumentar la cantidad de dinero que podemos destinar a una inversión.

**Cartera de inversión:** Combinación de activos financieros en los que tenemos depositado recursos monetarios con la idea de generar una plusvalía. Cuando se habla de aumentar o reducir el peso de un determinado activo en la cartera se refiere a aumentar o reducir los recursos monetarios depositados en ese activo frente al resto de los presentes en la cartera.

**Cotización/valor bursátil:** Precio de una acción o de un valor que cotiza en bolsa o en un mercado económico.

**Dividendo:** Parte de las ganancias de una sociedad que se distribuye periódicamente entre sus accionistas.

**Índice bursátil:** Es un valor numérico, que se calcula según los precios de mercado de cada uno de los valores que lo componen en un momento determinado. Los índices se crean con valores individuales que generalmente tienen en común pertenecer al mismo sector o a un mismo país.

**Inversión/Desinversión:** Depositar/retirar recursos monetarios de un activo financiero.

**Mercado financiero:** Espacio (físico, virtual o ambos) en el que se realizan intercambios de activos financieros definiéndose así sus precios.

**Ponerse largo/corto:** Ponerse largo se usa para referirse al hecho de comprar un activo financiero esperando que su valor aumente. Ponerse corto es el caso contrario, vender un activo financiero (que se toma prestado) esperando que su precio se reduzca para después recomprarlo, devolverlo y ganar la diferencia en el precio.

**Precio de cierre:** Precio al que se ha realizado el último intercambio de un determinado activo financiero antes del cierre de la sesión en un mercado financiero.

**Tendencia de mercado (alcista/bajista):** Tendencia en cual un mercado se mueve en una dirección particular durante un intervalo de tiempo (alcista si el activo sube de precio, bajista en caso contrario). Esta definición de tendencia se puede extrapolar a la de un activo concreto. Se dice que dicho activo corrige cuando se produce un cambio en la tendencia.

## Anexos

### English Summary

#### Abstract

This paper studies the applicability of Evolutionary Computation to financial risk management. Particularly, the problem of how to invest the available capital in a way that optimizes the probability of dealing with all expected negative cash flows. A tool is developed using evolutionary strategies, which in a simplified way, solves the problem. Given certain cash flows and available financial assets, the algorithm finds, based on the historical data of these financial assets, the optimal combination of them to reduce the liquidity risk. This paper describes the tool and the results of the applied tests.

**Key words:** Artificial Intelligence. Evolutionary Computation. Financial Risks. Investment. Liquidity.

## Introduction and goals

Artificial Intelligence's presence is raising in every sector. The increase in productivity due to the application of AI is unquestionable. According to a report developed by Accenture, capital and labour, so far, the engines of economic growth, are taking a back seat to AI, which is transforming the way to generate growth [1].

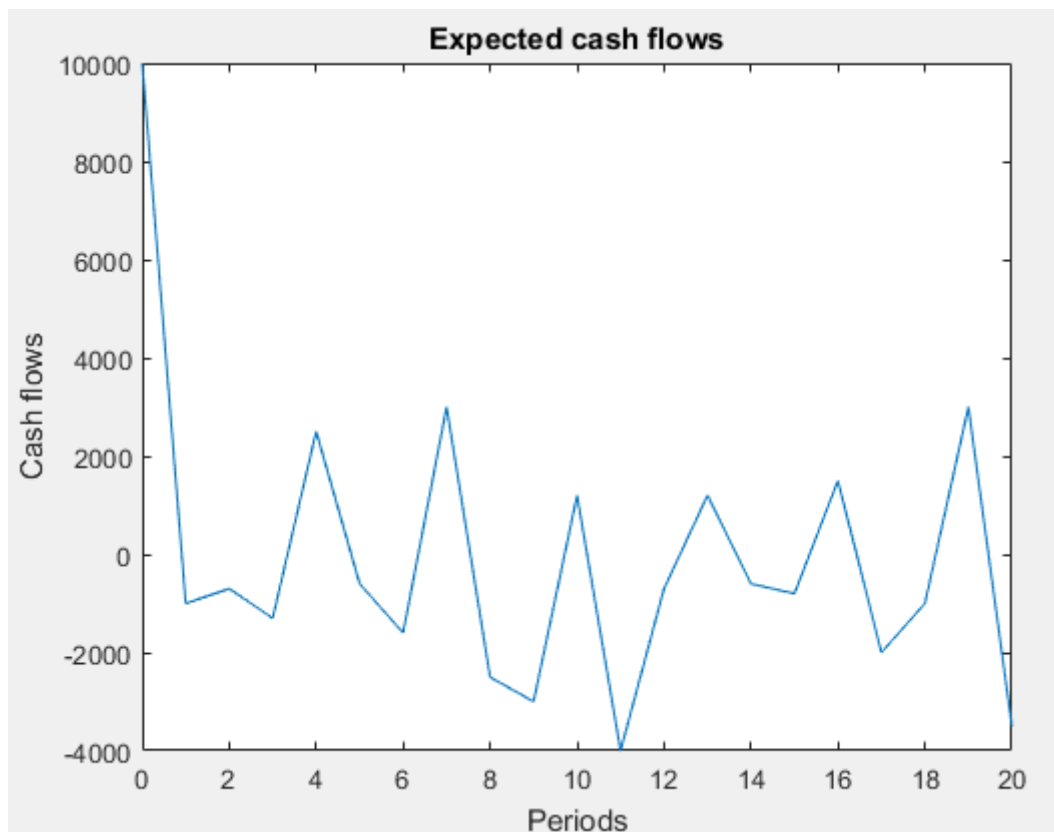
One of the areas affected by this transformation that is provoking AI is the finances. The constant emergence of new Fintechs (a name given to companies dedicated to the use of information technology to offer financial services) is an example of this. Financial institutions and other companies in the sector must adapt quickly to new technologies to maintain their position in the market. Some examples of the transformation that is provoking the AI in the sector are the new IA-based fraud detection systems, banking chatbots or algorithmic trading.

This paper shows the usefulness and value that could be provided by AI, in particular evolutionary computation, in the field of financial risk management, developing a tool with this goal.

A system of allocation of financial resources based on liquidity requirements is developed. However, it is a simplified implementation, that is, a proof of concept to verify the possibility of obtaining satisfactory results applying this branch of the AI to this goal. It is not intended to develop a fully functional tool for professional use.

The problem underlying the development of this tool is how a company can manage its financial resources to minimize liquidity risk, that is, in which financial assets distribute their capital to maximize the possibility of being able to cope with all payments. This problem is the engine for the development of the tool, however, it can be used with different approaches in the field of finance, not exclusively with the mentioned one. An example that is the personal financial planning. The approach in this case would be how to manage the savings to achieve a certain goal, for example: if I have a monthly saving capacity of X euros, how can I invest them to have a Y amount in 30 years and that allows me to retire. In that case, every month would have a positive cash flow of X and finally a negative cash flow of Y. The system would recommend a distribution of the capital in financial assets that would guarantee in a greater number of cases to be able to enjoy that quantity Y on the date set.

In a real scenario, to carry out a professional execution would require a very complete database with a historical data of very different financial assets: debt of different countries, debt of companies in different markets, shares of companies from different countries and dedicated to diverse activities, indexes of some markets, etc. The more information available, the better the results. A more careful study of cash flow forecasts would also have to be made, including the development of a more complete financial asset simulation system than the used in this proof of concept. However, the computational cost of the above would be very high and as already mentioned, the goal is not to develop a complete tool, but to test its usefulness, value and possibility of scale. For this we take some simplifications and assumptions that will be discussed in detail below.



**Figure 24.** Example of expected cash flows in a given case.

Figure 1 shows a graph with cash flows. It starts with an amount of 10,000 monetary units and over 20 periods money is entering (the graph is above 0) and money is coming out (if not). The goal consists, therefore, in the development of a tool that, given limited

financial assets (investment opportunities) and pre-established cash flow requirements, be able to define a combination of these financial assets that maximizes the probability of coping with all negative cash flows. That is, whenever there is a negative flow it must be capital enough to pay it. In the case of Figure 1 the negative cash flows have a higher value than the positive (considering the initial capital) therefore in this case if no profits are generated through the capital investment it isn't possible to make the payments.

The operation of the algorithm consists of: an exhaustive simulation of the possible future scenarios, that is, behaviours that the assets can have. To subsequently seek the combination of these assets that in a greater number of scenarios can meet the pre-established cash flows.

The objective of the work consists of two distinct objectives:

1. Analyse the applicability of evolutionary computing techniques in finance.
2. Perform a proof of concept of a tool that applying evolutionary computing techniques in support of financial risk management to provide value to its users.

These companies dedicated to the use of information technologies to offer financial services denominated fintech, are developing digital tools completely disruptive with the established business models, through the creation of new and more efficient ways to provide these services.

In the context of financial inclusion, fintech have great potential. Thanks to the competitiveness created by their new technologies developed tools, the old business models are being challenged and must adapt quickly since it is proving that financial services can be provided in a much faster, responsible and efficient way.

Access to financial products and services is more achievable than ever, especially for consumers living in rural regions lacking the structures of a modern economy. Fintech cannot simply make those products and services more accessible, they can also make it easier and more profitable to do business by reducing the cost of doing business.

Inclusion in finance is growing rapidly thanks to fintech. Some of the most significant advances they are making are: microfinance, electronic banking and loan applications enable faster lending decisions and allow lenders to serve more clients and in different locations simultaneously. International money transfers, thanks to fintech these are cheaper and faster. Access to credit histories, there is a lot of personal financial

information so far difficult to store and share, which translated into complicate access to credit, fintech facilitate the management of all this information. Electronic payments, etc.

Summarising, everything said seems to lead us towards a much more globalized economy. An economy in which technology will be much easier to access to financial services by anyone regardless of their place of birth. This will result in increased ease of doing business. Access to credit will be much simpler thanks to the greater ease of accessing credit histories, which together with the simplification of international transfers will result in an increase in entrepreneurship, which is the engine in generating prosperity.

The question below would be, how does the proposed tool fit within this disruptive context? From the perspective of this evolution to a more global and inclusive economy, this tool would collaborate in the management of financial risks efficiently in companies that cannot dedicate much capital to develop a competitive department in this aspect. On the other hand, it would be a support tool with a great contribution of value that helps in the decision-making of larger companies. Finally, focusing the tool from a supportive perspective in making personal investment decisions, could support large volumes of small investors at a lower price, thus helping to include more people in the financial services, not needing large capitals to hire a custom manager.

At the personal level, there are two main reasons that have led me to choose this topic as a thesis. In the first place, my motivations to develop a computational tool applied to finance. In a second place, why use evolutionary computing to develop this tool.

Firstly, as a double-degree student in Computer Science and Business Administration the training I have received has been hybrid. Finance is the part that interested me the most in the business part and computing in the computer science part. With this paper, I try to use artificial intelligence techniques with an approach in supporting financial decision making. Although it is an exclusive work of computer science, not a mixed one of both grades, I am motivated much more the development of a tool to which I see a practical usefulness in a field that I like as it is the finances, that a more theoretical development or applied in another field that I find less interesting.

Secondly, the choice to make use of evolutionary computing is motivated by the fact that in my curriculum (2011 plan) it is impossible to receive training in this subject. The curriculum leaves us very little free to configure subjects freely, allowing only to choose between the branch in Information Systems or Computing, not including the latter no

subject focused on evolutionary computing. For the mentions that were made on evolutionary computing in other more generic subjects such as Artificial Intelligence in organizations, it seemed to me a very interesting topic and to complete my training in this branch I decided to apply it in the development of my thesis in Computer Science.

On the other hand, the socio-economic context has been another factor that has propitiated my choice of this topic. As described previously, artificial intelligence is generating very significant changes affecting every aspect of life. The inclusion of these technologies is provoking an unprecedented increase in productivity, which translates into an increase in wealth that will change the way we see the world globally. So, what better way to contribute to this evolution towards an economy that is ever faster, more efficient and globalized than participating in the development of tools with this objective?

## Summary

In the first place, it is important to emphasize that in this paper a proof of concept is made, that is, an attempt is made to show that Evolutionary Computation can be applied in the field of ALM (asset and liability management) obtaining valuable results. It is not intended to develop a fully functional tool for professional use, therefore, the problem is a simplification of a real professional case.

For the development of the tool some simplifications and assumptions are taken, which facilitate its development and especially reduce its computational cost. These limitations imposed on the tool make it testable without affecting its scalability. The limitations of the tool are:

- Full capital invested. It is assumed that initially all the available capital will be invested and subsequently all the fruit of the invested capital plus the positive cash flows from which they are enjoyed will be reinvested following the structure of the portfolio. This is very unlikely to happen in a real case, such this kind of decisions should be made according to the client, taking each project as a different case and giving a personal treatment. Each client should choose between which margins of liquidity he wants to keep and the tool could recommend staying out



of the market, or not invest 100% of the capital in certain situations for being more prudent.

- Reinvestment according to a fixed capital structure. It is assumed that the capital structure of the portfolio remains fixed over time. At the initial time, different percentages of capital would be used to buy certain financial assets, each of them at a price determined by the market at that moment. Subsequently, following the natural evolution of the market, their prices would vary, causing the initially fixed weights in each asset to vary, with the assets with a better behaviour having a higher weight and those with a worse behaviour having a lower weight in the portfolio. The simplification that is made for proof of concept is to treat the capital structure of the portfolio as if it were kept fixed constantly. This simplification would be equivalent to assuming that at the end of each period it is divested and re-invested (the discounted amount plus the positive or negative flow of that period) according to the initial capital structure. In practice, although it also depends on the duration of periods, this would not be practical, since the cost of disinvestment and reinvestment operations could constantly undermine the benefits of investments.
- Only long strategy. It is only allowed to take long positions in the assets, that is, to buy them. You cannot "bet" that the price of these will fall to gain that margin of fall. Short positions, that is, selling an asset without holding it hoping that it can be repurchased shortly at a lower price to repay the underlying, require different capital guarantees that make them be treated separately. As in the previous cases, it is a personal decision that should be treated on a case-by-case basis, but to simplify the operation of the algorithm, only long positions are allowed in the proof of concept.
- Only 10 financial assets. As mentioned previously, the ideal to obtain the best results from this tool would be to have a very complete database, which would have different financial assets such as debt, stocks and indices of very varied markets. However, this factor significantly affects the computational cost of the algorithm, so in this work it has been decided to simplify and use only 10 financial assets. As in previous cases, this is easily scalable and could increase the number of assets available without major complications.
- Without considering the cost of the operation or taxes. The costs imposed by the broker and the taxes are a drag on the profitability. In a real scenario, it would be

very important to take them into account, however, these costs are not considered when developing the tool. However, it would not be complicated to add them and in case of professionalizing the tool should be necessary to adjust it according to the regulatory framework of each region of operation and according to the costs of the broker with which it is operated.

- No limit of companies in the portfolio. For some reasons, such as commission costs, it is unlikely that an investor wants to have a very large number of financial assets in his portfolio. However, encouraged by the two assumptions discussed above, the algorithm will not have a limit of companies in the portfolio.
- Monthly periods. In the tool, monthly periods are used, that is, the cash flows are considered monthly, the portfolio returns are also calculated monthly and, therefore, as mentioned in an earlier point, the restructuring of the portfolio to maintain the initial fixed structure will also be made every month.
- Threshold of 20 months. For the proof of concept, we will use cases in which a forecast of the monthly cash flows is available for the next 20 months.
- Asset behaviour simulation using a stochastic model. For the correct functioning of the tool it is necessary to make predictions of the behaviour of the assets in the future. These predictions are made based on their past behaviour, in the test they are carried out using a SDE (Stochastic Differential Equation) model. However, in an advanced version of the tool it might be an option to make these predictions of the future behaviour of the assets using other tools, for example, based on neural networks.

Once the assumptions and simplifications have been described, the problem to be solved could be summarized as: given an initial available money amount and cash flows (20), what combination of the 10 financial assets given is most likely to cope to all negative cash flows? Understanding to face a negative cash flow that when this amount is discounted of the value at that moment of the portfolio, the value of this one is still positive.

The tool developed is, as discussed in previous sections, an evolutionary algorithm. The individuals that make up the population, that is, the candidate solutions, are possible compositions of the investment portfolio. In our simplified case, it will be a 10-position vector, in which each position represents an asset. At each position of the vector a real

number in the range  $[0,1]$  will be stored, and the sum of all vector values is exactly equal to 1. This number represents the weight of the asset over the total of the portfolio.

The fitness or adaptation function is used to measure how well a candidate solution fits the environment, that is, to quantify and distinguish a better solution from another. Also, the selection of parent individuals to generate the later generations is based on this adaptation function. In the tool, the adaptability to the environment is measured according to the probability of dealing with all the negative cash flows. To determine the fitness value of each candidate solution a Monte Carlo method is followed. This method gives us approximate solutions by simulating random tests repeatedly. In other words, many simulations are performed using a SDE (Stochastic Differential Equation) model with a random component. Then, we will face each candidate solution to a certain number of possible scenarios (understanding scenario as the sum of the simulations of all the assets); the fitness function will be the percentage of those scenarios in which the candidate solution can cope with all negative cash flows.

The selection of parent individuals is performed through a tournament selection mechanism. The size of the tournament and other parameters such as the probability of mutation are not pre-set and it is possible to regulate them in different tests to find the combination that best results. However, it is not the objective of this paper to find the optimal combination.

Different genetic operators of reproduction are used. In the first place, a cross between two parents is performed with a predetermined probability in each execution. The crossing that is made is a 1-point crossover. We randomly select a position between the second and penultimate, in our case between positions 2 and 9 (1 and 8 if we start counting by 0, in the language used, Matlab, it starts from 1). Next, the first part of the parent vector, which will contain the first position and a length between 1 and 9, will be joined with the second part of the mother vector, which will contain its last position and will have a length of  $10 - \text{length of the first part of the parent vector}$ , giving rise to a child vector of length 10. The same process will then be performed by changing the father and the mother, that is, by taking the first part of the mother vector and the second part of the parent vector. To do this the algorithm proceeds as follows: first perform a tournament and select an individual parent. With a probability  $p$  of crossing defined before the execution makes another turn or and then selects a single parent. Then makes a cross, the resulting vector will have his first half of the father and the second of the mother. If the

new population that is being generated has not yet reached its maximum size (also pre-set before the execution) considering the new child individual, then a second child is generated who will have his / her first half as the mother individual and second as the father. With a  $1-p$  probability, a second individual will not be selected and therefore no crossover will be performed, the child will go to the next generation identical to the father, for now, since a mutation operation is performed later.

Subsequently, on the resulting child(s), a mutation operation will be performed. This is to modify each entry of the child vector with a  $q$  pre-set probability before execution. In case of mutation the input in question is replaced by a random value between 0 and 1.

Finally, the last operator used is a type of mutation, a permutation. That with a probability  $r$  any position of the child vector will be interchanged with any other. This operator has special meaning in the tool due to the nature of this, i.e. if we do not forget that our individual is a vector in which each position represents a financial asset, it seems a good way to mutate the exchange within a portfolio in which the rest remains constant the weight in it of two assets.

As a last step, once the next generation is created, the portfolio must be normalized, i.e. to make the sum of all positions in the vector be 1. It should not be forgotten that our individuals represent 100% of an investment portfolio, it would be absurd if the sum were other than 1. This would imply leverage, which is not allow. The way to readjust the values of the portfolio so that its sum is 1 is simple, each position is divided by the sum of all of them.

The process therefore of the algorithm would result in:

1. Initialize all variables that will determine a case. Cash flows are established, the historical data of the financial assets to be used are loaded, the stopping condition is determined (number of generations), the size of the population is determined, the number of scenarios / simulations to be used to evaluate the fitness value of each individual in each generation, the size of the tournament is established and the probabilities of crossing, mutation and permutation defined above are determined.
2. An initial random population of the pre-set size in the previous step is generated. In other words, an array with as many rows as the size of the population is created

and as many columns as historical financial assets have been entered. It is normalized so that the sum of all the positions of each row is 1.

3. The fitness of each generated individual is evaluated. This is done by performing as many asset behaviour simulations as they were predetermined in step 1 and by checking how many of them individuals could cope with all negative cash flows. This is done by multiplying the value of the portfolio before a period by the increments of each asset that forms the portfolio in that period weighted according to the structure of the same. After this is added the value of the cash flow of the period, if the result is negative would be equivalent to the company could not face the payments, and so on with all simulated scenarios to finally obtain a fitness value of each individual.
4. If the stop condition happens, the algorithm would stop and show the individual with the best adaptation value, that is, show the composition of the portfolio and your likelihood of dealing with all cash flows in any scenario. If the stop condition is not given, the next generation will be created.
5. The best individual of the current generation would go directly to the next one and then they would be selected by tournament parents who, after applying the genetic operators described above (crossing, mutation and permutation), would go on to form the next generation. When the new population had the size determined in step 1 it would normalize and re-run 3-4-5 until the stop condition is fulfilled.

## Conclusions and future work

The tool, with all the limitations and simplifications assumed, has been tested with different cases and works correctly. Therefore, it could be concluded that the work has satisfactorily fulfilled its goal.

However, the results have been somewhat biased by the limitations implemented. They fulfil their stated objective with the description of the problem. Cash flows were considered, which, except in one case, imply that they could not meet negative cash flows. The algorithm has been shown to find investment portfolios that would save this situation and improve them over generations until finding an optimal value.

The tool is functional, therefore, as a proof of concept the conclusion is satisfactory. However, to use the tool in a professional environment it would have to take into account certain factors and work on them.

1. First, the supply of financial assets must increase. Instead of working with stocks of companies directly, it would probably be more efficient to work with indexes. Either these geographic indexes should at least be given the option of investing in US, European and Asian indexes, or indices by sectors. On the other hand, debt products, both public and private, and with several geographic areas would be necessary.
2. Another feature that incorporates the rebalancing of the portfolio over time. It is more complicated face all the negative cash flows with a fixed portfolio determined at time 0, especially while the larger sea of the time predicted in the cash flows, than with a constant updatable portfolio.
3. Simulation of asset behaviour with more elaborate models that make use of artificial intelligence could be tested. For example, if we use networks of neurons for the prediction of asset behaviour and these prove to predict better the results obtained would be more precious. It should not be forgotten that the results obtained are based on a simulation of many scenarios, the more accurate this simulation the better the results.
4. In this test, the parameters haven't been too optimized to obtain better results, partly because the characteristics of the assets and historical used led to the algorithm converge quickly. In real cases, the optimization of parameters should be treated in a personalized way, with the characteristics of the genetic algorithm being understood as parameters, but also the historical data used since its depth greatly affects the simulations.
5. Of course, eliminate the simplifications assumed for proof of concept. How to allow long and short whenever the customer agrees or allow to keep part of the capital out of the market.